This version: 14/10/02 11:52

A Flexible Tool for Model Building: the Relevant Transformation of the Inputs Network Approach (RETINA)

Teodosio Pérez-Amaral,
Departamento de Analisis Economico
Universidad Complutense de Madrid,
teodosio@ccee.ucm.es

Giampiero M. Gallo,
Dipartimento di Statistica "G.Parenti"
University of Florence, Italy
gallog@ds.unifi.it

Halbert White,
Department of Economics
University of California, San Diego,
hwhite@weber.ucsd.edu

ABSTRACT

A new method, called relevant transformation of the inputs network approach (RETINA) is proposed as a tool for model building and selection. It is designed to improve some of the shortcomings of neural networks.

It has the flexibility of neural network models, the concavity of the likelihood in the weights of the usual likelihood models, and the ability to identify a parsimonious set of attributes that are likely to be relevant for predicting out of sample outcomes.

RETINA expands the range of models by considering transformations of the original inputs; splits the sample in three disjoint subsamples, sorts the candidate regressors by a saliency feature, chooses the models in subsample 1, uses subsample 2 for parameter estimation and subsample 3 for cross-validation. It is modular, can be used as a data exploratory tool and is computationally feasible in personal computers.

In tests on simulated data, it achieves high rates of successes when the sample size or the R^2 are large enough. As our experiments show, it is superior to alternative procedures such as the non negative garrote and forward and backward stepwise regression.

1. Introduction

Model building and selection are crucial in statistical analysis and at some point in the effort, a decision must be made as of which among several specifications (possibly belonging to different classes of models) should be chosen to represent a relationship between a dependent variable and other variables of interest. Among these, one may prefer a parametric specification (either linear or nonlinear) where some interpretation of parameter values may be retained, or else suggest the adoption of flexible functional forms where the relationship among the variables is guided by other criteria of explanatory power. In this respect, the search for flexibility may be guided by the inadequacy of a linear model with Gaussian errors to represent data in a suitable way. Generalized Linear Models (McCullogh and Nelder, 1989) and Generalized Additive Models (Hastie and Tibshirani, 1990) for specific classes of problems and Artificial Neural Networks (White, 1989) provide leading examples of such a strategy.

Within each class of models, the problem of selecting the specification is far from trivial. Approaches to model selection are numerous: not only do they differ between one another, but they present peculiarities which reveal the importance given by each to different aspects of modelling itself.

Some methods focus on the relationship between a model and its interpretability according to some theory, others are based on hypothesis testing between competing models; some judge upon the trade-off between explanatory power and parsimony in the retained specification, others are based on the performance of a model in explaining a set of data not used for estimation especially when the flexibility of the in-sample specification may lead to overparameterization; and so on. One popular approach in econometrics the so-called general-to-specific methodology whereby from a specification with a certain degree of complexity we would seek more parsimonious representations of the data which retain the same information in a simpler form**.** For a recent debate on this approach and its capability to recover the traits of the DGP, see Hoover and Perez, 2000, and the discussion contained therein, especially the somewhat skeptical view by Granger and Timmerman. This method involves a battery of diagnostic tests on estimated coefficients and residuals to achieve that.

No approach is perfect, especially when misspecification of a model relative to the process which generated the data is always a possibility; and hence all approaches to model selection have their own flaws. Hypothesis testing in the area of model choice is

reputed as potentially dangerous (cf. Granger et al., 1995) given the implicit advantage attributed to the model under the null hypothesis in a nested framework or the possible ambiguity of results in a non-nested context. Moreover, one of the undesirable aspects of such an approach is the need to resort to pairwise comparisons.

In frameworks in which a penalty function for the number of parameters weighs on the value of the likelihood function to provide a number which can be used to select a model (as in the Akaike's AIC or the Schwartz's BIC) there is always the issue of which form such a function should take given some undesirable properties of such information criteria to systematically choose over- or under-parameterized models in some circumstances.

Model selection based on out-of-sample performance is also prone to problems and, in fact, after the pioneering work by Granger and Newbold in the early 70s (Granger and Newbold, 1973), it has become standard practice only in recent years to adopt testing procedures for predictive ability whereby some measure of performance (such as the Mean Squared Prediction Error, but again the choice of the criterion is not neutral) is used in a formal hypothesis testing framework (cf. Diebold and Mariano, 1995, West, 1996, White, 2000).

In general, researchers are aware that the activity of model selection through a specification search (led, for example, by a forecasting performance criterion) may translate into the choice of a good model just due to luck: the pervasive investigation of the same data either individually or collectively may distort the view on the "right" model to work with.

In this paper we present a tool that may be useful for model building and selection: we suggest a novel approach to investigating a data structure with the purpose of achieving a flexible and parsimonious representation of the mean of a variable, conditional on a set of variables deemed of interest for the phenomenon at hand. Our approach may prove useful to investigate phenomena where one can think of a (large) list of variables potentially informative in describing the conditional mean (behavior) of a variable, but s/he does not have any strong priors as of the form of the relevant function, or of the relevance of individual variables for the data at hand**.** The procedure may be even more useful when the data generating process involves a relatively small number of relatively large parameters. Customer credit scoring and demand for telecommunication services by firms using individual data are just two examples of such phenomena.

This approach, called the Relevant Transformation of the Inputs Network Approach (RETINA) is based on earlier work by White (1998) and has the flexibility of neural

network models in that it accommodates non-linearities and interaction effects (through non-linear transformations of the potentially useful variables in the conditioning set), the concavity of the likelihood in the weights of the usual likelihood models (which avoids numerical complexity in estimation), and the ability to straightforwardly identify a set of attributes that are likely to be truly valuable for predicting performance evaluation outcomes (which responds to a principle of parsimony). Moreover, it is computationally not demanding and has good finite sample properties. Even when compared to the initial suggestion by White (1998), RETINA has higher rates of success and better finite sample properties at a slightly higher computational cost.

When selecting the relevant inputs, the approach has some elements of similarity to the subset regression literature in statistics (Miller, 1990). To mark the differences, we will report the results of comparisons of RETINA to some of these methods, namely, stepwise regression – where some variables are eliminated according to the significance of their parameters - and nonnegative Garrote – where some parameters are set to zero while others are shrunk toward zero (Breiman, 1995).

In performing the selection, our approach relies on a cross-validation scheme which is aimed at limiting the possibilities that a good performance is due to sheer luck. In particular, we design a division of the observations on three homogeneous sub-samples and a selection procedure where possible models are estimated in the first sub-sample and their performance evaluated in the second sub-sample by means of (out-of-sample) mean squared prediction error. After a “candidate” is selected this way, a similar procedure is repeated, this time estimating various models derived from the “candidate” in the second sub-sample and cross-validating them on the third sub-sample by means of an information criterion. We are not providing a theoretical justification for the division of the overall sample in three sub-samples or for the choice of two different measures for evaluation purposes, except, heuristically, the evidence of a good performance of the procedure in the various simulations we performed. While we are sure that worse choices exist, we are uncertain about the existence of better criteria for specific types of DGP’s which would further improve the procedure.

There are several aspects that our approach does not address: first and foremost, we are aware of the fact that any model specification exercise is intended to be one of finding a good approximation according to some criteria and not one of finding the “true” model. Second, in the simulations here we treat a case in which the data generating process (DGP) is i.i.d., although extensions to heterogeneous and/or dependent processes

(including the important case of non stationary variables) along these lines can be envisaged. Third, since we are not concerned about retrieving the form of the function linking the variables in the conditioning set to the dependent variable.

This approach does not solve the problem of choosing which class of models is best suited to represent certain data (e.g., in a time series context, linear, bilinear, ARCH, Threshold Autoregressive, and so on) assuming that the true DGP is among them.

The structure of the paper is as follows: in section 2 we define the RETINA and we justify the adopted steps in the procedure. In section 3 we give a brief description of the main differences to similar existing approaches. Section 4 contains a description of the simulation design where we have envisaged a number of situations in which the DGP may contain some elements of noise for model selection (presence of outliers, of structural breaks, of sparse data, etc.). We limit the presentation of the results to a few leading cases transferring the main bulk of the detailed results to an Appendix. Concluding remarks follow.

2. The RETINA procedure.

As mentioned in the introduction, the tool presented here, the RElevant Transformation of the Inputs Network Approach (RETINA) shares some characteristics of earlier work by White (1998), in that it has the flexibility of neural network models, the computation simplicity of the usual likelihood-based methods for which the likelihood function is concave in the weights, and the ability to straightforwardly identify a parsimonious set of attributes that are likely to be truly valuable for predicting performance evaluation outcomes by way of model selection criterion based on a cross-validation scheme. The “relevant-input” network model described below is computationally efficient so that it can be used in desktop computers and has good finite sample properties. RETINA has higher rates of success and better finite sample properties at a slightly higher computational cost than the original proposal by White (1998).

Let us start by considering a (large) number of variables potentially of interest in describing the behavior of the mean of a dependent variable Y. Given the lack of information on the form of such a relationship, in order to maintain a degree of flexibility one may want to use a nonlinear transformation of the input variables, say $Z = z(X)$. In pursuing these transformations, we will keep in mind our goal of identifying a parsimonious set of (transformed) attributes that are likely to be truly relevant for predicting out of sample

outcomes for Y. Hence, we need to be careful, that the transformations we choose are not highly correlated with one another, as highly correlated transforms will not provide a great deal of independent predictive information.

Concavity in the parameters may be achieved by allowing the effects of the Z's on Y to be exerted in a linear fashion, providing a model of the form:

$E(Y/X) \approx z(X)' b$ in the regression case or, more in general

$E(Y/X) \approx F(z(X)' b)$ where F is a suitable link function (e.g. the logistic cdf for bynary classification problems) . We will rule out the appearance of new parameters inside z because that may result in non-concavity.

An important feature of White (1998) which we will exploit here is to avoid the evaluation of all the 2^m possible models when we have m candidate regressors in the set of transformed variables Z, and then applying some form of model selection. Rather, in the present formulation, the approach envisages the selection of a number (of order proportional to m) of candidate models, inserting new explanatory variables on the basis of relevance (for instance, ranking the candidate regressors according to their correlation in absolute value with the dependent variable). At the same time the degree of dependency of the new information added is controlled for, by keeping the amount of collinearity among the regressors under a threshold parameter λ chosen by the experimenter ($\lambda \to 0$ – new regressors approach orthogonality; $\lambda \to 1$ – new regressors are highly collinear).

As with all flexible modelling, the issue then becomes one of not favoring the model which performs the best in sample, in order to avoid overparameterization. This is achieved here in two steps: after having estimated the models in a first sub-sample, they are cross-validated in a second subsample, then the best model re-estimated and a check adopted as of whether a different choice of the order in which the regressors are inserted one by one in the model would lead to a more parsimonious representation. Thus, very important features of the procedure are that it uses disjoint sub-samples for cross-validation and an out-of-sample forecast ability as the criterion for model selection.

2.1 The procedure in detail

The procedure can be described as follows. Assume that for each individual observation i, (i=1, …, n) we observe a value of the response variable Y_i and we have available candidate predictor attributes X_{ih}, h = 1, …, k, where k is potentially a very large number.

Then the following steps are performed:

i. From the original attributes X_{ij} form a collection W_{ij}, j = 1, …, m of transforms of the original attributes. For example, include in the collection of transforms the original X_{ij}´s (the "level 0 transforms"), all of their squares and cross products, and all of their inverses and cross-ratios (taking care to avoid perfect multicollinearity and divisions by zero). Call this collection the "level 1 transforms". If desired, a set of "level 2 transforms" can be formed by appending the level 1 transforms to the original level 0 transforms and then taking level 1 transforms again. The process can be continued to any desired level, but to keep things simple, we further discuss only the level 1 transforms. For simplicity and concreteness, we have discussed using transforms of the form $X_{ih}{}^{\alpha} X_{ij}^{\beta}$, $\alpha, \beta = -1, 0, 1$, but there are many other possibilities that could be used instead.

These transforms W_{ij} fulfill our requirements of providing a rich set of univariate predictors that embody not only nonlinearities but also interactions. To extract a relatively parsimonious subset that may provide a useful basis for predicting Y_i, we proceed as follows:

ii. Divide the sample in three disjoint homogenous subsamples.

iii. In subsample 1, compute a relevance measure of the relationship between Y_i and W_{ij}, for example the sample correlation $\hat{r}_j$, j = 1, …, m.

iv. Rank the predictors in order of (the absolute value of) the saliency measure, say, $|\hat{r}_j|$ of subsample 1, and denote the ranked W_{ij}'s as $W_{i(j)}$, j = 1, …, m, where $W_{i(1)}$ has the highest **(**absolute) saliency with Y_i and $W_{i(m)}$ the lowest.

v. Create a candidate subset of predictors. First, include $W_{i(1)}$ in the candidate subset, then proceed through the ranked list of $W_{i(j)}$, including in the predictor attribute set any $W_{i(j)}$ for which the R^2 of the regression of that $W_{i(j)}$ on the current set of included predictor attributes is below λ, where λ is a prespecified threshold value, $0 \quad \lambda \quad 1$, and excluding $W_{i(j)}$ otherwise. Denote the resulting set of transforms $z_l(X_i)$. The smaller the correlation the candidate predictors have with each other, the better, other things being equal. By making the selection depend on λ, we control this correlation. By repeating our candidate variable selection process for a grid of values for λ, say $\lambda_1, \ldots, \lambda_\nu$, we obtain corresponding candidate transformations $z_l = z_{ll}$, $l = 1, \ldots, \nu$. We can then select a best choice for λ from the collection $\{\lambda_1, \ldots, \lambda_\nu\}$

in a manner analogous to the way in which a best choice for the number of hidden units q is selected for the single layer feedforward network model. Hence, for each $\zeta_l(X_i)$, we estimate the model in subsample 1 and we compute an out of sample prediction criterion (e.g.: the cross-validated mean square prediction error) in subsample 2.

vi. Choose the candidate model (and λ) that optimizes the criterion in subsample 2.

vii. Estimate the parameters of the candidate model in subsample 2. By estimating in subsample 2 we get essentially unbiased estimates of the parameters.

viii. Use the estimates from subsample 2 to compute a measure of the out of sample forecast ability in subsample 3. The recommended model is the submodel with the best out of sample forecast ability (e.g. lowest mean square prediction error) in subsample 3.

As a matter of fact, under point viii. above, we use a somewhat more elaborate method for arriving at the final set of predictors that takes advantage of the fact that the elements of each z_l have a natural ordering in terms of their univariate relevance with the target variable in subsample 1. Because of this ordering, one can proceed in a step-wise fashion for a given value of λ_l: one estimates the submodel in subsample 2 including only the first element of z_l, then including the first two elements of z_l and so on.

For each submodel estimated, one computes the forecast criterion in subsample 3 and chooses the one with the best forecast criterion (e.g.: AIC). This criterion is then compared across different values of λ to select λ^*. This may permit the selection of a more parsimonious model than one would get by ignoring the natural ordering in the elements of each z_l.

In practice we repeat point viii. resorting the elements of z_l by their univariate saliency with Y in subsample 2. This may allow the consideration of a wider range of candidate models.

Points iii. through ix. above can be repeated changing the order of the subsamples. At this point, we may have more than one candidate model. The recommended model (and optimal λ) would be the one that has the best performance in an appropriately defined sense when estimated in the whole sample.

The data generating process may not be within the class of models considered by the researcher. However s/he may use a parametric model of some aspect of the

phenomenon that is the “preferred model”, a useful approximation for a particular purpose, e.g.: estimation of a conditional mean, hypothesis testing or out of sample forecasting.

The “preferred model”, in general, may be different depending on its intended use and the data available. Economists some times use different models of consumption depending on whether they need it for estimation of a given parameter, choosing among competing theories or performing out of sample forecasts. The type of data available also influences the choice of model, e.g.: cross section, time series or panel data. The level of aggregation is also relevant for the choice of model: individual, family, city, region or country levels are some examples.

RETINA´s recommended model should be taken as a suggestion for a useful approximation to some unknown relationship. The researcher needs to assess its coherence and the rationale for the suggested transformations. S/he can add variables, delete others or introduce restrictions based on prior knowledge, theoretical or empirical considerations.

The heuristic justification for using three disjoint subsamples is as follows. We want disjoint subsamples so that the information and the statistics we compute are independent across samples. We use the first subsample for model selection, i.e.: choosing which transformations look promising in terms of the saliency feature in subsample 1. Then we estimate the models in subsample 1 and cross validate them in subsample 2. The parameters and standard deviations estimated in this fashion are biased away from zero as shown by Miller (1990). Therefore we need to estimate the parameters of this model in an essentially unbiased way and we do that by estimating them using the second subsample. Once we have estimated the parameters appropriately, we proceed to check the out of sample performance of the model by calculating a measure such as the MSEP using fresh data; in the third subsample[1]. This description suggests that using two subsamples is not enough: we do not have fresh data either for the estimation or for the cross validation. Using more than three is unnecessary, since we do not need additional data sets after the third subsample.

[1] Shao (1993) considers the selection of a model with the best predictive ability. He uses a leave-n_v-out cross validation, which is consistent when n_v, the number of observations reserved for validation satisfies n_v /n 1.

3. Related approaches.

Here we comment on the relationships between RETINA and other model building and selection tools such as neural networks, White (1988), stepwise regression, Miller(1990), the London School of Economics methodology, Hoover and Pérez (2000) and non-negative garrote, Breiman (1995). We also comment on the relationship with other types of models such as generalised linear models and generalised additive models, Hastie and Tibshirani (1990). We do not comment on other model building methods such as ACE, Breiman and Friedman (1985) and CART, Breiman et al. (1984) and Chipman et al. (1998), because they are no closely related to RETINA.

RETINA is designed to overcome some of the drawbacks of **neural networks**, White (1988). To that end, it uses ingredients that were previously available in the literature. We point out some similarities and differences of the RETINA procedure with previous procedures.

Our procedure has some common features with neural networks models, such as the flexibility, which is afforded here using nonlinear transformations of the inputs while maintaining linearity in the parameters within the link function. On the other hand, neural networks models achieve flexibility by allowing nonlinearities in the parameters. With respect to the objective function, RETINA uses an out of sample predictive criterion, while neural nets use an in-sample goodness of fit criterion.

RETINA also has features in common with **stepwise regression**, Miller (1990), e.g.: the ability to search for a subset of relevant regressors, in a non-exahustive fashion. In particular, RETINA performs a selective search guided by a saliency feature of the regressors. A difference between the two procedures is that RETINA perfoms a search based on an out of sample criterion, while stepwise regression uses an in sample model selection criterion.

The London School of Economics (**LSE**), Hoover and Pérez (2000), from general-to-specific approach to model building and selection starts with a reasonably general specification of a model and through parameters and residuals tests selects a parsimonious model that adequately represents the relationship under consideration, RETINA can be considered a from-general-to-more-general-to-specific methodology. First, it expands the range of possible regressors by including the transformations of the inputs, then it considers models that include both the inputs and their transforms, then it narrows the search to the most promising models using a selective search criterion (saliency). One

important difference is that the LSE methodology uses in sample tests while RETINA uses an out of sample criterion.

An alternative model building and selection approach is **non negative garrote**, Breiman (1995). It is a method for doing subset regression. It starts with a linear regression including all the possible explanatory variables and selects subsets by zeroing and/or shrinking coefficient estimates. It works well in experimental data compared to subset selection when there is a small number of large coefficients. Non negative garrote uses a **cross validation** as the model selection criterion and does not consider explicitly the transformations of the original inputs.

Generalised linear models and generalised additive models, Hastie and Tibshirani, (1990), are models linear in the parameters. Like our procedure, they can incorporate nonlinear link functions. Since RETINA considers in addition models that involve interactions of the original inputs, the models considered by RETINA are broader than generalised linear and generalised additive models.

As we have shown, several ingredients of RETINA were already present in the literature. Some of them have well established roots, like the generalised linear models, the out of sample forecasting criteria and the selective search. The use of the parameter for controlling collinearity, this particular saliency feature and the division in three subsamples may be less widespread. All of them are simple and have some intuitive appeal.

4. Simulations.

Because analytic results are difficult to come by in this area, the major proving ground is testing on simulated data. We explore the finite sample properties of RETINA and compare it with backward stepwise regression (Miller, 1990) and non-negative garrote (Breiman, 1995).

We compare it with backward stepwise regression because it is a widely used model selection procedure. The reasons for comparing it with non negative garrote are that it is focused on the forecast ability of the model, is more stable than subset regression and is superior both to subset selection and ridge regression when the number of relevant regressors is small (like the present situation). However, non negative garrote is computationally demanding, compared to our procedure.

We investigate how well does RETINA select the right model in the following sense:

1. With what frequency does RETINA choose a model that coincides with the data generation process (DGP) when the DGP is within the candidate models considered? (including irrelevant X's). We consider both cases, when the DGP is linear in the X´s and when the DGP is nonlinear in the X´s.
2. How does the procedure perform when the DGP includes discrete explanatory variables?
3. How does RETINA perform when the DGP includes X's with sparse data?
4. How sensitive is it to the presence of outliers in the DGP?
5. How sensitive is it to the presence of a structural break in the DGP?
6. How fine a grid for λ, (parameter that controls for collinearity) do we need?

4.1. Design of the experiments.

The data were generated using the data generation process (DGP):

DGP1: $y_i = \alpha_0 + \alpha_1 x_{1i} + \alpha_2 x_{2i} + EW\, u_i$, $\quad i = 1, \ldots, n$,

where $\alpha_0 = \alpha_1 = \alpha_2 = 1$, x_{1i} and x_{2i} are jointly normal with correlations $\rho = 0.5$ or 0.9. The error term u_i is iid N(0,1), EW is a parameter that controls the standard deviation of the error, to obtain average values of the R^2 for each experiment around 0.75, 0.50 and 0.25 respectively. The sample sizes are n=100, 200 and 1000. Other sample sizes were used with similar results and are not reported here. The number of replications for each run of each experiment was 1000.

DGP2: $y_i = \alpha_0 + \alpha_1 x_{1i} / x_{2i} + EW\, u_i$, where the second term depends on the ratio x_{1i} / x_{2i} and everything else is as in DGP1 except that $\rho = 0.5$. Analogously,

DGP3: $y_i = \alpha_0 + \alpha_1 x_{1i} x_{2i} + EW\, u_i$,

For the case of a discrete explanatory variable we use the same setup and parameter values as DGP1 except that now x_{1i}' is a dummy variable that takes the value 1 with probability 0.5 and 0 otherwise:

DGP4: $y_i = \alpha_0 + \alpha_1 x_{1i}' + \alpha_2 x_{2i} + EW\, u_i$.

For the case of sparse data we use the same setup as in DGP1, except that x_{1i}'' is iid N(0,1) with probability 0.2 and zero otherwise.

DGP5: $y_i = \alpha_0 + \alpha_1 x_{1i}'' + \alpha_2 x_{2i} + EW\, u_i$.

To check for the sensitivity to outliers we go back to DGP1 and use u_i' which is the same as u_i except that when the absolute value of u_i is larger than 1.96 standard deviations, it is multiplied by 5 (and alternatively 2.5 or 10). That is, we expect 5% outliers.
DGP6: $y_i = \alpha_0 + \alpha_1 x_{1i} + \alpha_2 x_{2i} + EW\ u_i'$.

We explore the ability of RETINA to recover the right regressors when a structural break in the parameters has occurred. The DGP is linear, as in DGP1 but now
DGP7: $y_i = \alpha_0 + \alpha^*_1 x_{1i} + \alpha^*_2 x_{2i} + EW\ u_i$,
where $\alpha^*_1 = \alpha^*_2 = 1$ for the first half of the sample and $\alpha^*_1 = 0.5$, $\alpha^*_2 = 2$ for the second half.

4.2. Results.

For the simulations we used a program written in GAUSS. For Experiment 1, the data were generated using DGP1 with $\rho = 0.5$. RETINA was used with level one transforms of x_{1i} , x_{2i} and x_{3i}, and the constant, where x_{3i} is an irrelevant regressor which is also jointly normal with the same distribution and correlations as above. The parameter λ varies from 0 to1 by increments of 0.1. The maximum number of candidate regressors (W_{ij}'s) is 25, of which only three are relevant. The total number of possible candidate models to consider is 2^{24}, since the constant is always in the candidate model. RETINA evaluates around 2x24 different candidate models.

We count a success when RETINA chooses a candidate model which coincides with the DGP. The percentages of successes and two standard deviations (represented by the vertical bars at each point) are displayed in Figure 1a. This suggests that if we have either a large R^2 or sample size, the percentage of successes is close to 100%.

Figure 1a. Dgp1, linear case, r=0.5

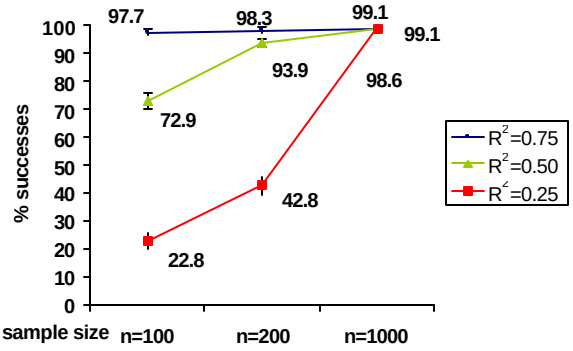


Table 1. Percentages of successes of RETINA for different DGP´S and R^2 =0.5

	Sample size		
DGP	n = 100	n = 200	n = 1000
DGP1 linear	73	94	99
DGP2 ratio	73	82	95
DGP3 product	96	99	99
DGP4 dummy	43	73	96
DGP5 sparse data	-	33	93
DGP6 outliers	65	93	99
DGP7 struct. break	39	67	99

Table 2. Percentages of successes of RETINA versus other procedures for DGP1 and R^2=0.5

R^2 = 0,5	Sample size		
DGP1	n = 100	n = 200	n = 1000
RETINA	73	94	99
Non-negative garrote	5	11	54
Stepwise regression	9	8	9
Simpler RETINA	60	68	76

Table 1 present an overview of the results of the use of RETINA with different DGP´S. For simplicity, we round up the percentages to the closest integer and show only the leading case of R^2 = 0.5. See more details in the Appendix. RETINA works well when the DGP includes transformations of the inputs, discrete explanatory variables, sparse data, outliers or structural breaks. These results suggest that when the DGP is among the models considered by RETINA, there is a high probability, in some cases close to one, that it is recovered. The probability of success increases with the sample size. Graph 1 and the results in the Appendix also suggest that this probability increases with R^2.

These experiments suggest that RETINA meets a necessary condition for its usefulness as a model selection strategy, since, it recovers the DGP with high frequency when the DGP is within the candidate models.

Table 2 above shows the performance of RETINA vis a vis its competitors, such as non negative garrote, backward stepwise regression and a simpler version of RETINA: without resorting the three subsamples, and without degrees of freeedom corrections. The simulation results suggest that RETINA outperforms its rivals when the objective is to recover the DGP. This is another necessary condition for the usefulness of the procedure. See more details in the Appendix.

The grid for λ, the parameter that controls collinearity, need not be too fine. In the experiments, using λ between 0 and 1 by increments of 0.1 is fine enough. This limits the computational costs of the procedure.

4.3. Comments.

RETINA may be a useful tool for model building and selection. It can be used as a data **exploratory tool** to suggest possible models and transformations of the inputs.

RETINA is a **modular procedure**. One can substitute some ingredients, e.g.: use different levels of transformations, logarithms, a different saliency feature or cross validation criteria, and the procedure works in a modified fashion.

RETINA can consider a **maximum of 2^m-1** models, where m is the total number of candidate regressors, that is, the columns of W when level one transforms are applied. For instance, when we have a constant and two varying inputs: X={1, x_1, x_2} and W={1, x_1, x_2, x_1x_2, x_1^2, x_2^2, x_1^{-1}, x_2^{-1}, $x_1 x_2^{-1}$, $x_1^{-1}x_2$, x_1^{-2}, x_2^{-2}, $x_1^{-1}x_2^{-1}$}. The potential number of models is $2^{13}-1$= 8191. If we always include a constant, the potential number of models is 2^{12} = 4096. For a constant and three varying inputs the potential number of models, if they always include a constant, is 2^{24}= 16,777,216 models. A major advantage of a selective search (guided by a saliency feature) is to reduce the number of models actually evaluated.

RETINA allows for **likelihood-type estimation** techniques other than regression, and for the use of dependent observations. It may be more appropriate for cases in which there are a **few large nonzero parameters**, while other methods such as ridge regression may be more appropriate when there are many nonzero, but possibly small, parameters (Breiman, 1995).

5. Concluding remarks.

A new method, based on RIPNET of White (1998), called relevant transformation of the inputs network approach, RETINA, is proposed for model building and selection. It is designed to have the flexibility of neural network models, the concavity of the likelihood in the weights of the usual likelihood models and the ability to identify a parsimonious set of attributes that are likely to be relevant for predicting performance evaluation outcomes.

The procedures splits the sample in three disjoint subsamples and uses subsample 1 essentially for model selection, subsample 2 for cross validation and parameter estimation and subsample 3 for cross validation.

To assess the finite sample performance of RETINA we performed simulations in which we record the ability of the procedure to recover the DGP. In general, the results are encouraging, except when the sample size or the R^2 are small. RETINA, seems to perform well with DGP's linear and nonlinear in the inputs, dummies, sparse data, outliers and structural breaks.

The procedure is computationally feasible in personal computers and the rates of success are better than some competing criteria, such as non negative garrote and backward stepwise regression. RETINA can be used as an exploratory tool, is modular and flexible and the models can be easily modified by the user. This suggests that RETINA can be useful for applied researchers. The present version of RETINA is applicable to independent identically distributed observations. It may be worth considering the applicability of RETINA to other types of data, such as time series and panel data.

References.

Akaike, H. (1973) “Information Theory and an Extension of the Likelihood Principle”, in B. N. Petrov and F. Csaki (eds.), *Proceedings of the Second International Symposium on Information Theory*. Budapest: Akademiai Kiado, 267-281.

Breiman, L. (1992) “The Little Bootstrap and Other Methods for Dimensionality Selection in Regression: X-fixed Prediction Error,” Journal of the American Statistical Association, Vol. 87, No. 419, 738-754.

Breiman, L. (1995) “Better Subset Regression Using the Nonnegative Garrote” Technometrics, Vol. 37, 4, 373-384.

Breiman, L. and H. Friedman (1985) “Estimating Optimal Transformations for Multiple Regression and Correlation” Journal of the American Statistical Association, Vol. 80, No. 391, 580-619.

Breiman, L., Friedman, H., Olshen, R. and C. Stone (1984) *Classification and Regression Trees,* Wadsworh Statistics/Probability Series, Belmont, California.

Burnham, K. and D. Anderson (1998) *Model Selection and Inference: A Practical Information-Theoretic Approach*, Springer-Verlag, New York.

Chipman, H., E. George and R. McCulloch (1998) “Bayesian CART Model Search”, Journal of the American Statistical Association, Vol. 93, 935-960.

Diebold, Francis X., and Roberto S. Mariano, Comparing Predictive Accuracy, Journal of Business and Economic Statistics, v.13, no.3 (July 1995), pp. 253-63.

Granger, C.W.J., M. King and H. White, (1995) , Comments on Testing Economic Theories and the Use of Model Selection Criteria, Journal of Econometrics, 67, 173-187.

Hastie, T. J. and R. J. Tibshirani (1990) *Generalized Additive Models*, Monographs on Statistics and Applied Probability 43, Chapman and Hall, London.

Hoover, K. and J. Perez (1999) “Data Mining Reconsidered: Encompassing and the General-to-Specific Approach to Specification Search”, manuscript, Department of Economics UC Davis.

Miller, A. J. (1990) *Subset Selection in Regression*, Monographs on Statistics and Applied Probability 40, Chapman and Hall, London.

Shao, J. (1993) “Linear Model Selection by Cross-Validation”, Journal of the American Statistical Association, Vol. 88, No. 422, 486-494.

Shao, J. (1996) “Bootstrap Model Selection”, Journal of the American Statistical Association, Vol. 91, No. 434, 655-665.

White, H., (1989), Learning in artificial neural networks: a statistical perspective, Neural Computation, 1, pp. 425--464.

White, H. (1998) *Artificial Neural Network and Alternative Methods for Assessing Naval Readiness*. Technical Report, NRDA, San Diego.

White, H. (2000), "A Reality Check for Data Snooping," Econometrica, 68, 1097—1126.

West, K. D. (1996). Asymptotic inference about predictive ability. Econometrica 64, 1067–84.

Zhang, P. (1992) "On the Distributional Properties of Model Selection Criteria", Journal of the American Statistical Association, Vol. 87, No. 419, 732-737.

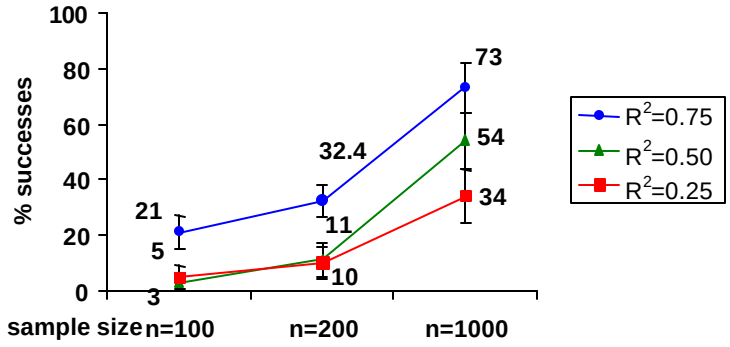
Appendix

We present the outcomes of the experiments with more detail.

1. DGP1. For comparison, with RETINA, we present in Figure 1b the results of Experiment 1b, with the same DGP and parameter values, but using Breiman's (1995) non-negative garrote instead of RETINA. We use tenfold cross validation, make available to non negative garrote all the Wij's and set to zero the coefficients whose estimated absolute values are below 0.01.

The rates of successes follow the same patterns as those of Figure 1a, increasing with the sample size and the R^2**,** however, they are uniformly lower than those of RETINA. The execution time of non-negative garrote is more than two hundred times that of RETINA, due to the nonlinear optimizations and the tenfold cross-validation used by this method. That is why the numbers of replications used for non-negative garrote are smaller. We have also limited the maximum value of s (the garrote parameter) to 6 or 4 for faster convergence and execution. On these grounds, RETINA is superior to non-negative garrote as model selection criterion.

Figure 1b. Dgp1, garrote, linear case, r=0.5, 100



In Experiment 2, we analyze the same DGP1 (including the same values of EW) as in Experiment 1. The main difference is that here we allow for more collinearity among the candidate regressors, $\rho = 0.9$ and therefore, the R^2 are higher than in Experiment 1. The results are summarized in Figure 2, which represents the % of successes of RETINA for recovering the regressors of the DGP. The vertical bars are two standard deviations. and suggest that collinearity may be damaging for model selection. The results seem good for high R^2 and large sample sizes, but are generally poor for samples of sizes 100 and 200. This suggests that collinearity may be damaging for model selection using this version of RETINA.

Figure 2. Dgp1, linear case, r= 0.9

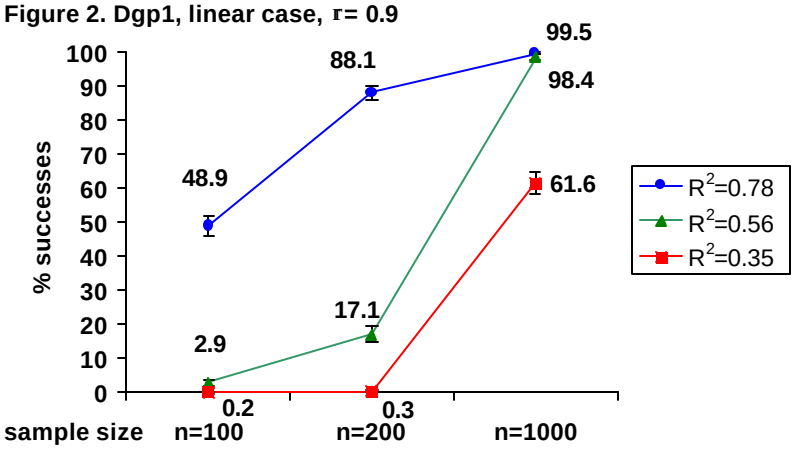
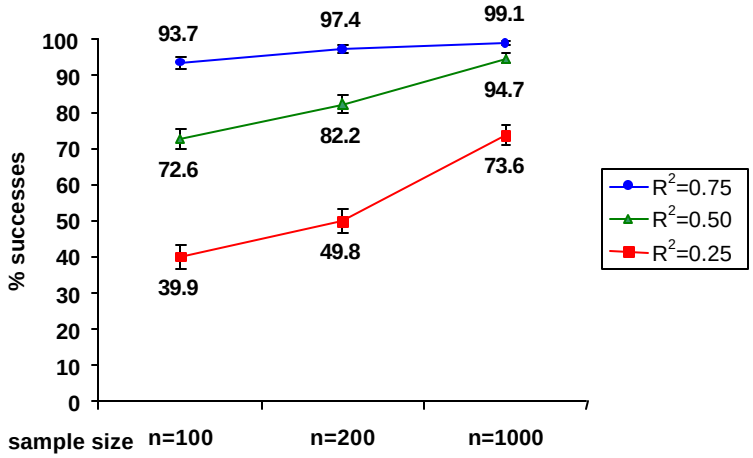


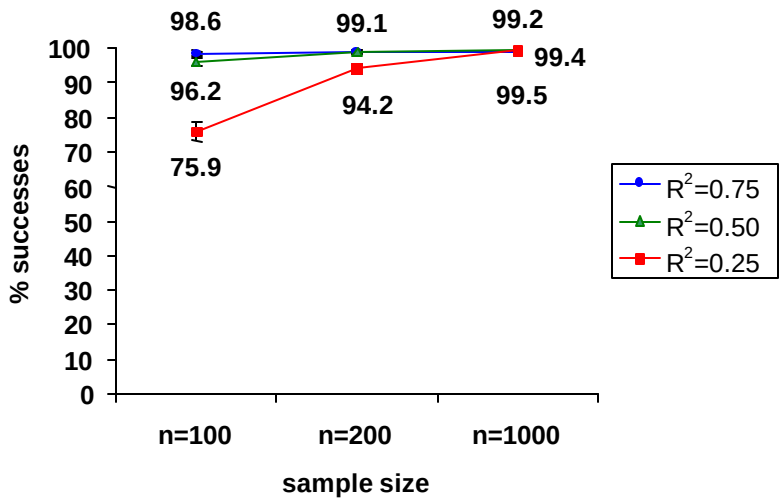
Figure 3. Dgp2: ratio x1/x2



In Experiment 3 we analyze the performance of RETINA when the data are generated in a nonlinear fashion by DGP2. The results are summarized in Figure 3. The rates of successes are again reasonably high if the R^2 or the sample size are large enough.

In the next Experiment we use the multiplicative DGP3. The results are summarized in Figure 4. In this case RETINA works well in all cases except for n=100 and

Figure 4. Dgp3, product x1*x2.



R^2=0.25.

. In the next Experiment, with DGP4, one of the regressors is a dummy variable that randomly takes the value 1 or 0 with probability 0.5. The results are summarized in Figure 5. Here, again, a large sample or a high R^2 are required for the procedure to yield a reasonable percentage of successes.

Figure 5. Dgp4, linear with dummy.

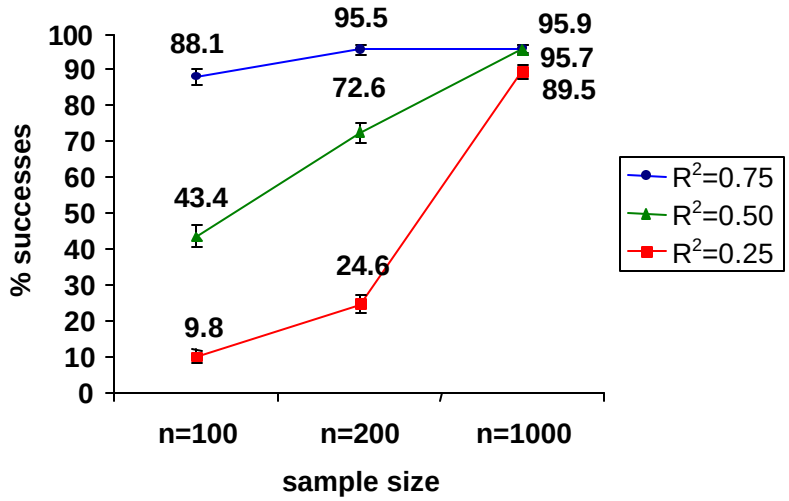


Figure 6. Dgp5, linear dgp, sparse data.

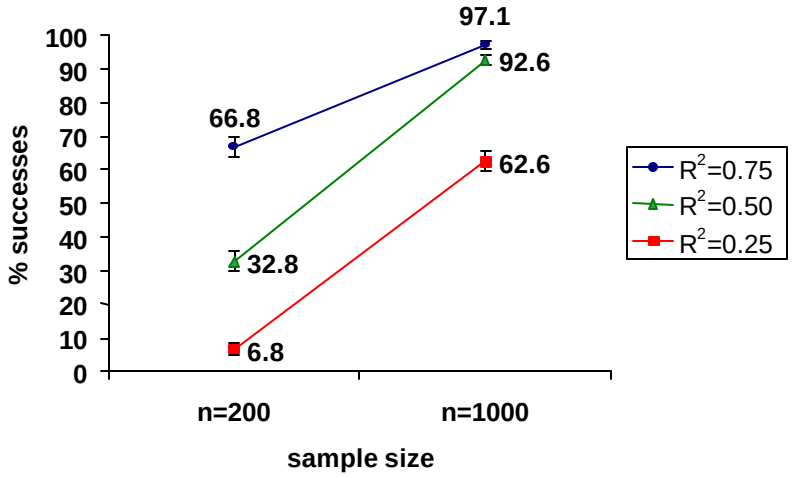
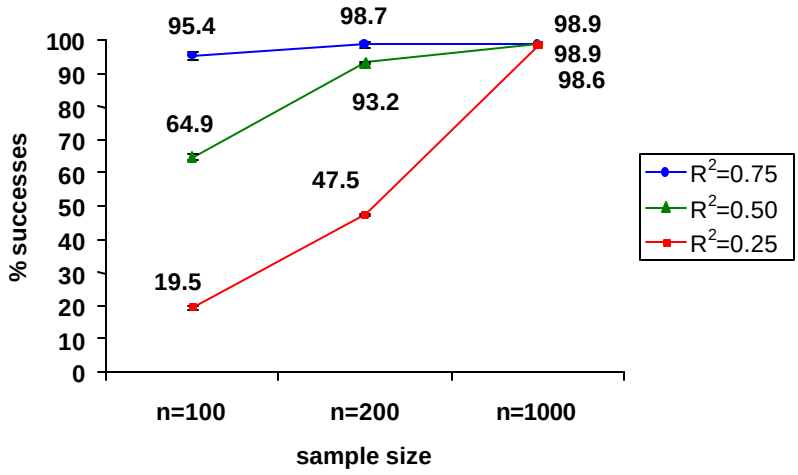


Figure 6 summarizes the results of applying RETINA to data generated by DGP5, that is, a linear model in which one of the regressors takes the value zero with probability 0.8 and is extracted from an iid N(0,1) otherwise. This is the case of sparse data. Here, no samples of size 100 were drawn because, frequently, after splitting the sample in three, all the observations for one variable in one of the subsamples were zero. If the sample size is large enough and the R^2 not too low the procedure works reasonably well.

Figure 7a summarizes the results of the experiments that use dpg6 to generate the data. Here, the values of the error term greater in absolute value than 1.96 are multiplied times 5. In this case, we do not expect RETINA to recover the dpg; instead we consider a success when it correctly identifies the regressors of the DGP. Again, when the sample size is large enough or the R^2 is high enough, the procedure can recover the regressors of the DGP with high frequency. Otherwise, it may fail to do so.

Figure 7a. Dgp4, linear with 5% outliers in error.



Next, we wanted to analyze how the presence of outliers affects the ability of RETINA to select the correct model. This is presented in Figure 7b. For that we used the n=100, R^2=0.50 case and use different values of the parameter that multiplies each of the outliers. For the last case of X10, each realization of the error larger in absolute value than 1.96 is multiplied times 10. The baseline case of no outliers is X1, for which the values of the error are not altered. The results suggest that RETINA may be quite robust to the presence of outliers. However, the estimates of the coefficients and their precision are adversely affected.

Figure 7b. Dgp4, linear, 5% outliers, varying their size

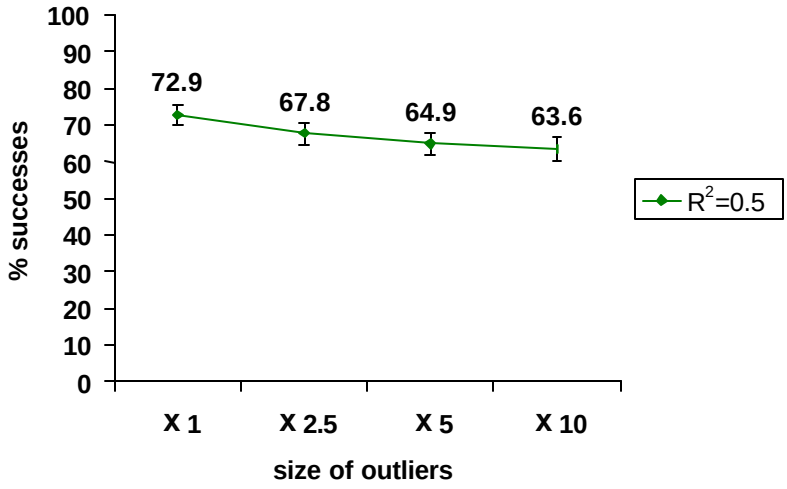
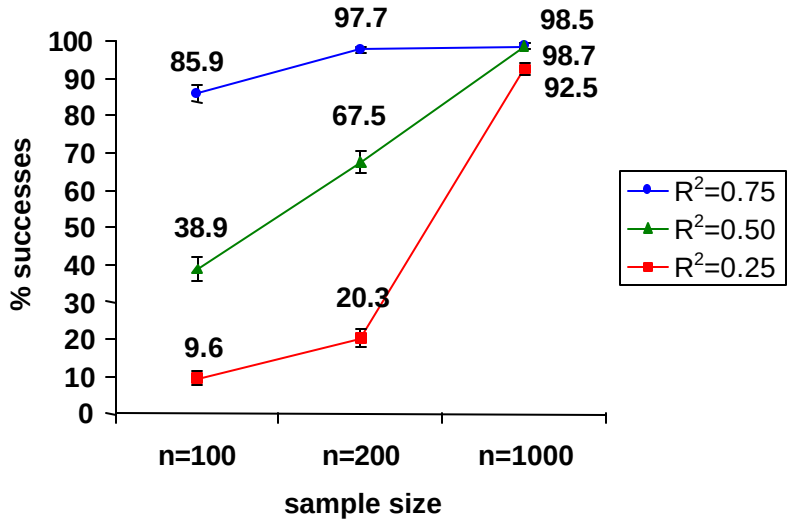


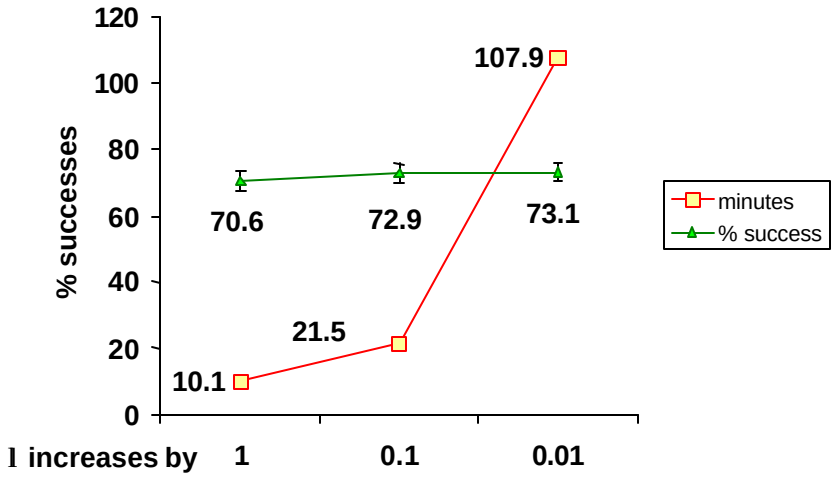
Figure 8 summarizes the results of RETINA when there is a structural break in the middle of the sample. In that case, it still achieves high rates of success for recovering the right regressors, except in the cases with low R^2 or relatively small samples. However it does not recover the DGP. Subsequent tests for structural change or residual analysis may detect the existence and location of the structural breaks.

Figure 8. Dgp7, linear, structural break.



In our experiments, a reasonable grid for the parameter λ is between 0 and 1 by increments of 0.1. A finer grid increases marginally the success rate while the execution time increases substantially. On the other hand, a less fine grid decreases somewhat the success rate while saving little computer time. This is suggested by Figure 9, in which we have repeated one run of Experiment 1, with DGP1, for R^2=0.50 and n=100 varying the increments of λ between 1 (only zero or one are considered), 0.1 and 0.01. In the Figure we show the rates of success and the minutes of execution for each of these values for the λ grid.

Figure 9. What grid for λ? (R^2=0.5, n=100)



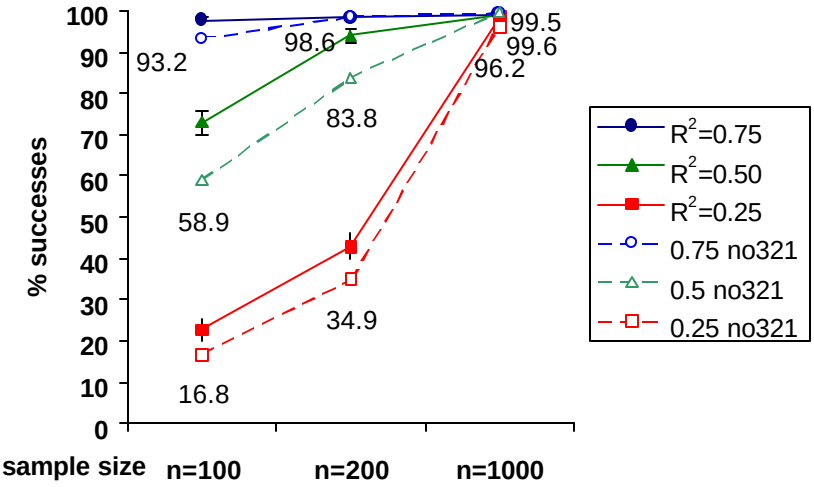
2. RETINA vs. RIPNET and simpler versions of RETINA.

In this section we compare RETINA with some simpler versions of RETINA and with RIPNET. In Table 1' we compare RETINA and a simpler version of RETINA, in which we do not perform the resorting of the three subsamples and therefore do not repeat the model selection procedure with the subsamples resorted as 321. Lines in dashes

Table 1', dgp1, linear case r=0.5 RETINA w/o 321

	n=100	n=200	n=1000
R^2=0.75	97.7	98.3	99.1
st dev	0.47	0.41	0.3
2 st dev	0.94	0.82	0.6
0.75 no321	93.2	98.6	99.5
R^2=0.50	72.9	93.9	99.1
st dev	1.41	0.76	0.3
2 st dev	2.82	1.52	0.6
0.5 no321	58.9	83.8	99.6
R^2=0.25	22.8	42.8	98.6
st dev	1.33	1.56	0.37
2 st dev	2.66	3.12	0.74
0.25 no321	16.8	34.9	96.2

Figure 1'. Dgp1, linear r=0.5, RETINA w/o 321



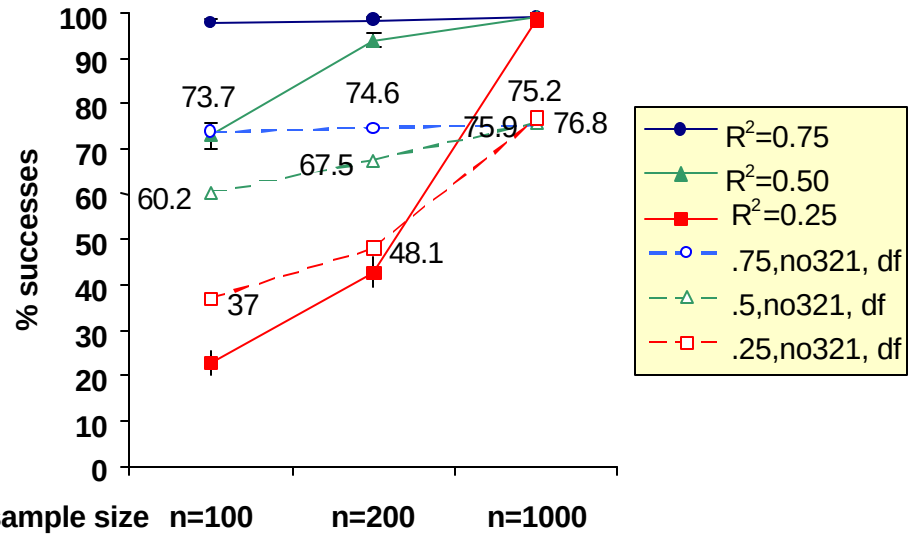
correspond to RETINA without 321 resorting, while solid lines correspond to RETINA.The percentages of successes are generally better for RETINA, suggesting that the 321 resorting increases the ability to select the DGP especially when R^2 and n are not large.

2.1. In Table 1'' we present the comparison of RETINA with a simpler version of RETINA which does not resort the subsamples as 321, and that does not use a degrees of freedom correction to compare the performance of the models that use different number of parameters.

Table 1", dgp1, linear case r=0.5 RETINA w/o 321 and w/o df

	n=100	n=200	n=1000
R^2=0.75	97.7	98.3	99.1
st dev	0.47	0.41	0.3
2 st dev	0.94	0.82	0.6
.75,no321, df.	73.7	74.6	75.2
R^2=0.50	72.9	93.9	99.1
st dev	1.41	0.76	0.3
2 st dev	2.82	1.52	0.6
.5,no321, df.	60.2	67.5	75.9
R^2=0.25	22.8	42.8	98.6
st dev	1.33	1.56	0.37
2 st dev	2.66	3.12	0.74
.25,no321, df.	37	48.1	76.8

Fig 1" Dgp1, linear r=0.5, RETINA w/o 321, w/o df



The elimination of the correction for degrees of freedom seems to hurt the performance of the procedure in most cases, especially when the R^2 is high or the sample size is large. When the sample size is large, the rates of success stabilize around 76% and do not approach 100%, as in RETINA.

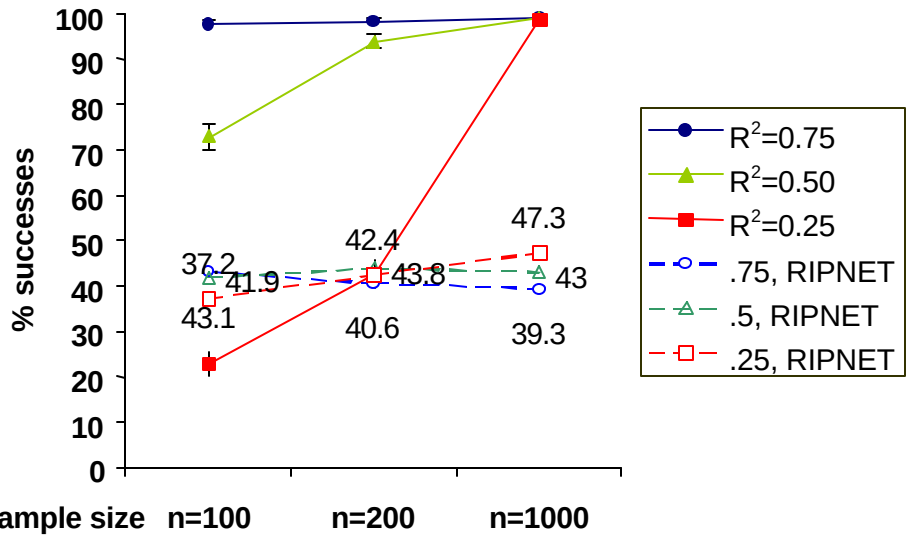
However, when the R^2 is low, the rate of success of the procedure without degrees of freedom correction is larger than the one of RETINA. This suggests that in this case, the degrees of freedom correction may lead to underparameterization.

2.2. In Table 1''' we present the first comparison of RETINA and RIPNET.

Table 1''', dgp1, linear r=0.5 RETINA and RIPNET

	n=100	n=200	n=1000
R^2=0.75	97.7	98.3	99.1
st dev	0.47	0.41	0.3
2 st dev	0.94	0.82	0.6
.75, RIPNET	43.1	40.6	39.3
R^2=0.50	72.9	93.9	99.1
st dev	1.41	0.76	0.3
2 st dev	2.82	1.52	0.6
.5, RIPNET	41.9	43.8	43
R^2=0.25	22.8	42.8	98.6
st dev	1.33	1.56	0.37
2 st dev	2.66	3.12	0.74
.25, RIPNET	37.2	42.4	47.3

Fig 1''' Dgp1, linear r=0.5, RIPNET



Average number of excess regressors in overparameterized models by RIPNET in Table 1'''. Dgp1, r=0.5.

	n=100	n=200	n=1000
R^2=0.75	2.92	2.77	2.77
R^2=0.50	2.93	2.64	2.67
R^2=0.25	3.04	2.71	2.61

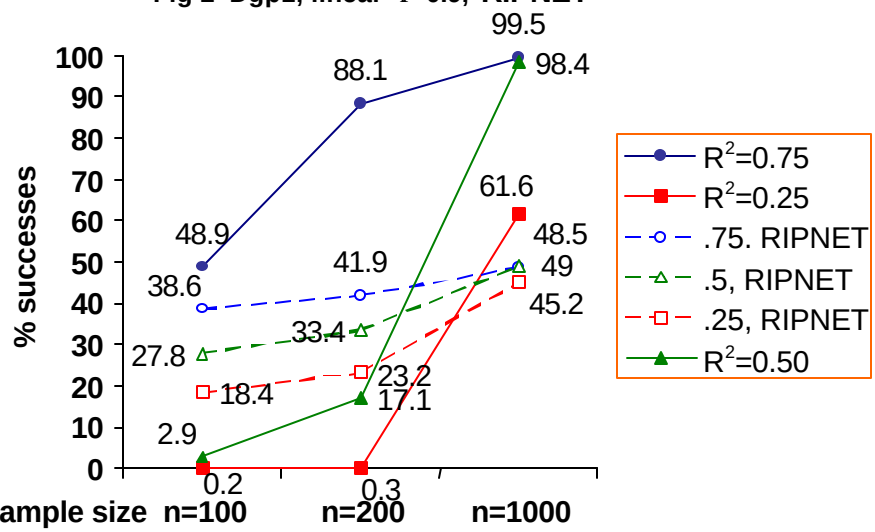
In this case, RETINA is better than RIPNET except for R^2=.25 and n=100. The rates of success of RIPNET are low, do not approach 100 when n increases and for R^2=.75 they decrease with n. The average number of excess regressors for RIPNET is between 2 and 3, which means that the model selected by RIPNET has on average around twice as many variables as the DGP. This shows no tendency to decrease with n or R^2.

2.3. RETINA vs. RIPNET, linear case and ρ=0.9.

Table 2', dgp1, linear case r=0.9 RIPNET

	n=100	n=200	n=1000
R^2=0.75	48,9	88.1	99.5
st dev	1.58	1.02	0.22
2 st dev	3.16	2.04	0.44
.75. RIPNET	38.6	41.9	48.5
R^2=0.50	2.9	17.1	98.4
st dev	0.53	1.19	0.4
2 st dev	1.06	2.38	0.8
.5. RIPNET	27.8	33.4	49
R^2=0.25	0.2	0.3	61.6
st dev	0.14	0.17	1.54
2 st dev	0.28	0.34	3.08
.25. RIPNET	18.4	23.2	45.2

Fig 2' Dgp1, linear r=0.9, RIPNET



Average number of excess regressors in overparameterized models by RIPNET in table 2'. Dgp1, r=0.9.

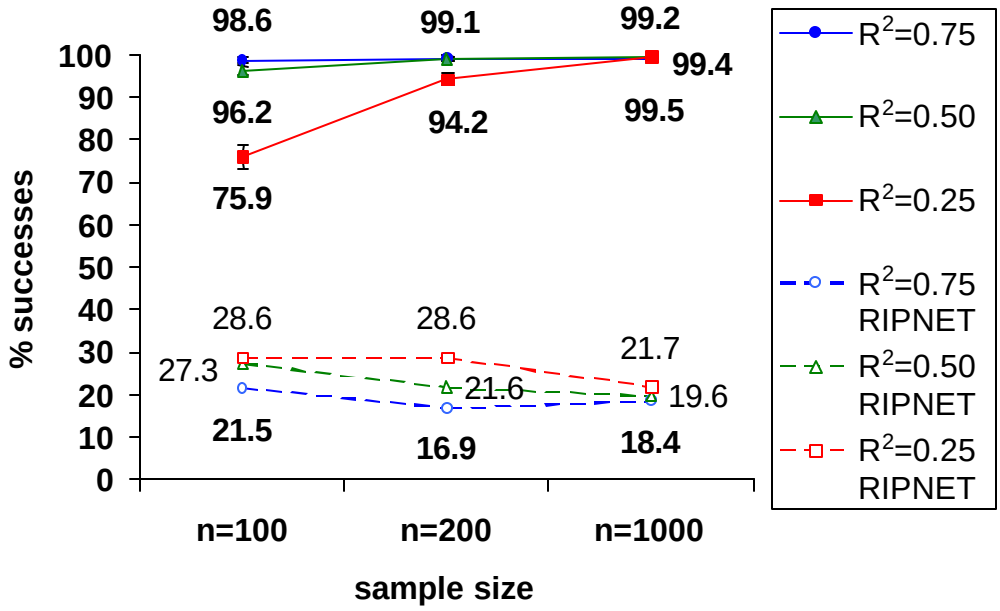
	n=100	n=200	n=1000
R^2=0.75	2.92	3.03	3.19
R^2=0.50	1.82	2.96	2.98
R^2=0.25	2.14	3.14	2.73

In this experiment the percentages of successes of RIPNET are often lower than those of RETINA and do not increase towards 100 with n. However, several are higher than RETINA, e.g. those for R^2=0.25 and 0.50 and n=100 and 200. This suggests the use of a criterion that employs a degrees of freedom correction less drastic than the one used by RETINA.

Table 4'. Dgp3, product x1*x2, RIPNET

	n=100	n=200	n=1000
R^2=0.75	98.6	99.1	99.2
	0.37	0.3	0.28
	0.74	0.6	0.56
R^2=0.75 RIPNET	21.5	16.9	18.4
R^2=0.50	96.2	99.1	99.4
	0.6	0.3	0.24
	1.2	0.6	0.48
R^2=0.50 RIPNET	27.3	21.6	19.6
R^2=0.25	75.9	94.2	99.5
	1.35	0.74	0.22
	2.7	1.48	0.44
R^2=0.25 RIPNET	28.6	28.6	21.7

Figure 4'. Dgp3, product x1*x2, RIPNET



Average # of excess regressors in overparameterized models by RIPNET in Table 4'. Dgp3, x1*x2,

	n=100	n=200	n=1000
R^2=0.75	2.65	2.73	2.89
R^2=0.50	2.66	2.67	2.88
R^2=0.25	2.9	2.58	2.87

2.4. RETINA vs. RIPNET when the DGP includes the product x_1*x_2. In this experiment RIPNET is uniformly worse than RETINA. The percentages of successes of RIPNET are low and remain low for large n. The rates of successes do not increase and

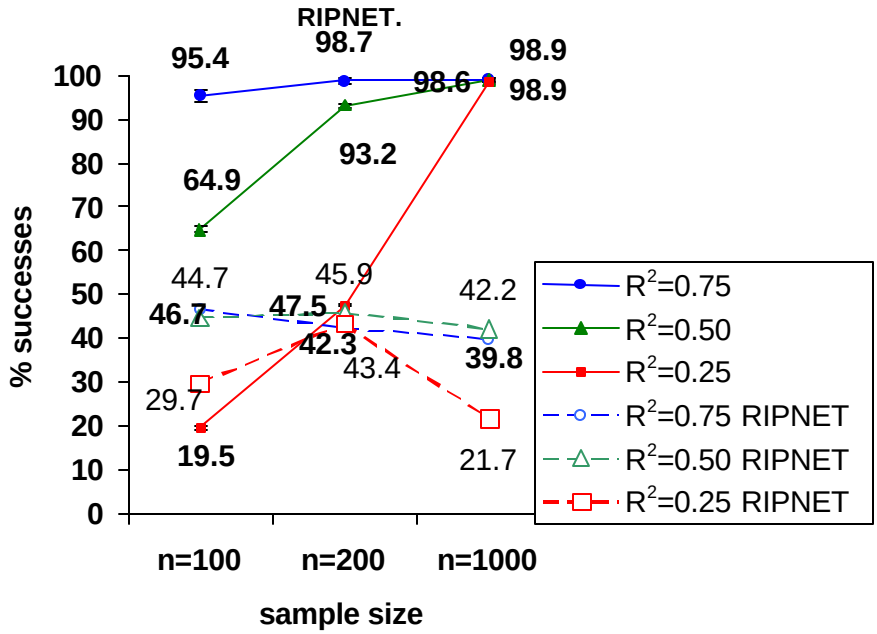
often decrease with n. RIPNET overparameterizes substantially. The DGP has 2 variables and the average number of variables in the models selected by RIPNET is between 4.5 and 4.9.

2.5. RETINA vs. RIPNET with outliers.

Table 7'. Dgp6, linear, 5% outliers in u, RETINA and RIPNET.

	n=100	n=200	n=1000
R^2=0.75	95.4	98.7	98.9
	0.66	0.36	0.33
	1.32	0.72	0.66
R^2=0.75 RIPNET	46.7	42.3	39.8
R^2=0.50	64.9	93.2	98.9
	1.51	0.8	0.33
	0.66	0.36	0.33
R^2=0.50 RIPNET	44.7	45.9	42.2
R^2=0.25	19.5	47.5	98.6
	1.25	1.58	0.37
	0.66	0.36	0.33
R^2=0.25 RIPNET	29.7	43.4	21.7

Figure 7'. Dgp4. linear with 5% outliers in u. RETINA and RIPNET.



Average number of excess regressors in overparameterized models by RIPNET in table 7'. Dgp4. 5% outliers.

	n=100	n=200	n=1000
R^2=0.75	2.93	3.14	3.21
R^2=0.50	3.51	2.94	3.03
R^2=0.25	3.68	3.05	2.87

In this experiment RIPNET is uniformly worse than RETINA, except for the case of R^2=.25 and n=100. The percentages of successes of RIPNET are low and remain low for large n. The rates of successes do not increase with n, and often decrease with n. In

summary, RETINA is generally superior to RIPNET, which has a strong tendency to overparameterize.

3. RETINA vs. stepwise regression. In this section we compare the performance of RETINA with stepwise regression. Stepwise regression is a popular model selection technique, implemented in some commonly used software packages. We follow Miller (1990, p. 48). Stepwise regression is often used to mean an algorithm proposed by Efroymson (1960), which is a variation on forward selection. After each variable (other than the first) is added to the set of selected variables, a test is made to see if any of the previously selected variables can be deleted without appreciably increasing the residual sum of squares. Efroymson's algorithm incorporates criteria for the addition and deletion of variables as follows.

a. Addition

Let RSS_p denote the residual sum of squares with p variables and a constant in the model. Suppose the smallest RSS which can be obtained by adding another variable to the present set is RSS_{p+1}. The ratio

$$R = \frac{RSS_p - RSS_{p+1}}{RSS_{p+1}/(n-p-2)}$$

is calculated and compared with and 'F-to-enter' value, say F_e. If R is greater than F_e the variable is added to the selected set.

b. Deletion

With p variables and a constant in the selected subset, let RSS_{p-1} be the smallest RSS which can be obtained after deleting any variable from the previously selected variables. The ratio

$$R = \frac{RSS_{p-1} - RSS_p}{RSS_p/(n-p-1)}$$

is calculated and compared with an 'F-to-delete (or drop)' value, say F_d. If R is less than F_d, the variable is deleted from the selected set.

The optimum F-to-enter for minimizing the mean square error of prediction in the case of random and correlated regressors, $F_e \leq 2n/n-p$, or a little less than 2 if n»p (Miller, p. 183). And the F-to-delete statistic has a value not greater than 1 (Miller, p. 207).

3.1 Linear case with ρ=0.5.

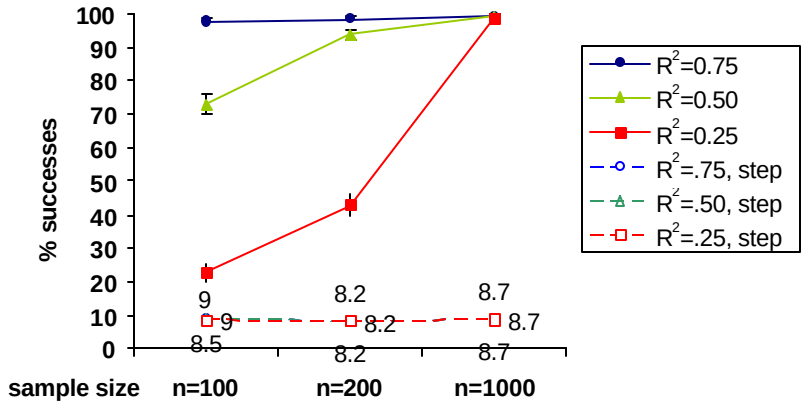
First we compare the performance of RETINA with stepwise regression in the same setting as Experiment 1 using DGP1 with correlations between regressors ρ=0.5. The data are the same both for RETINA and stepwise regression. All the transformations of the original regressors (W's) of RETINA are made available to stepwise.

Table 1S, Dgp1, r=0.5 RETINA vs. stepwise

	n=100	n=200	n=1000
R^2=0.75 RETINA	97.7	98.3	99.1
st dev	0.47	0.41	0.3
2 st dev	0.94	0.82	0.6
R^2=.75 step	9	8.2	8.7
R^2=0.50 RETINA	72.9	93.9	99.1
st dev	1.41	0.76	0.3
2 st dev	2.82	1.52	0.6
R^2=.50, step	9	8.2	8.7
R^2=0.25 RETINA	22.8	42.8	98.6
st dev	1.33	1.56	0.37
2 st dev	2.66	3.12	0.74
R^2=.25, step	8.5	8.2	8.7

Table 1S and Figure 1S (for stepwise) summarize the percentages of successes for RETINA and stepwise regression. The solid lines join the points corresponding to RETINA and the broken lines link those of stepwise. The percentages of success of RETINA are considerably higher than those of stepwise regression across sample sizes and coefficients of determination. Stepwise regression overparameterizes around 90% of the time, choosing models that on average have around 2.7 more parameters than the original 3 of the DGP.

Fig 1S Dgp1, r=0.5, RETINA vs. stepwise



Average number of excess regressors in overparameterized models by stepwise in Table 1S. Dgp1, r=0.5 and % overparameterizations.

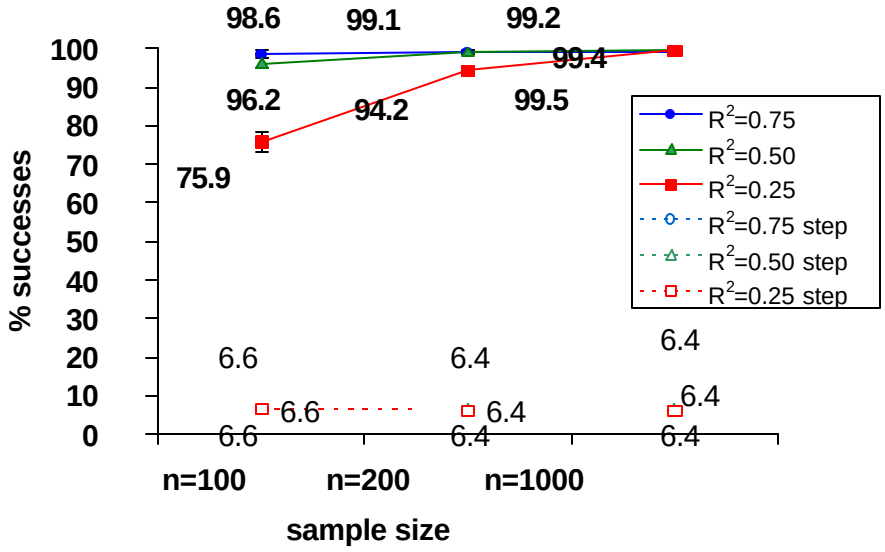
	n=100	n=200	n=1000
R^2=0.75	2.62	2.69	2.67
	91.00%	91.80%	91.30%
R^2=0.50	2.63	2.69	2.67
	90.70%	91.80%	91.30%
R^2=0.25	2.78	2.69	2.67
	83.50%	91.40%	91.30%

3.2 Linear case with r =0.9.

In this case we use DGP1 as before, but generate the x's with ρ=0.9. The results are summarized in Table 2S and Figure 2S. The percentages of success are generally higher for RETINA than for stepwise regression except for the cases of low R^2 and samples of 100 or 200. In general, there is a strong tendency of stepwise towards overparameterization, which occurs between 24 and 85% of the time, with an average number of excess regressors in overparameterized models between 5.51 and 2.43.

Table 2S, Dgp1, r=0.9 RETINA vs. stepwise

	n=100	n=200	n=1000
R^2=0.75	48.9	88.1	99.5
st dev	1.58	1.02	0.22
2 st dev	3.16	2.04	0.44
R^2=.75, step	14.9	13.7	14.2
R^2=0.50	2.9	17.1	98.4
st dev	0.53	1.19	0.4
2 st dev	1.06	2.38	0.8
R^2=.50, step	10.5	12.5	14.2
R^2=0.25	0.2	0.3	61.6
st dev	0.14	0.17	1.54
2 st dev	0.28	0.34	3.08
R^2=.25, step	4	6.1	14.2



Average number of excess regressors in overparameterized models by stepwise in Table 4S. Dgp3, product x1*x2, and % overparameterizations.

	n=100	n=200	n=1000
R^2=0.75	2.9	2.83	2.85
	93.40%	93.60%	93.60%
R^2=0.50	2.9	2.83	2.85
	93.40%	93.60%	93.60%
R^2=0.25	2.93	2.83	2.85
	92.10%	93.60%	93.60%

3.3 Nonlinear DGP3 with x_1*x_2.

In this experiment we compare the performance of RETINA and stepwise regression as model selection criteria. The data are generated by DGP3, that is a constant and the product of x_1 times x_2. The results are summarized in Table 4S and Figure 4S. The rates of success of stepwise regression are around 6.5 % while those of RETINA are always above 75%. Stepwise regression shows a strong tendency to overparameterize which

Table 4S. Dgp3, product x_1*x_2, RETINA vs. stepwise

	n=100	n=200	n=1000
R^2=0.75 RETINA	98.6	99.1	99.2
	0.37	0.3	0.28
	0.74	0.6	0.56
R^2=0.75 step	21.5	16.9	18.4
R^2=0.50 RETINA	96.2	99.1	99.4
	0.6	0.3	0.24
	1.2	0.6	0.48
R^2=0.50 step	27.3	21.6	19.6
R^2=0.25 RETINA	75.9	94.2	99.5
	1.35	0.74	0.22
	2.7	1.48	0.44
R^2=0.25 step	28.6	28.6	21.7

occurs over 92% of the time. When it overparameterizes, stepwise uses on average between 2.93 and 2.83 extra regressors when only 2 of them belong in DGP3.

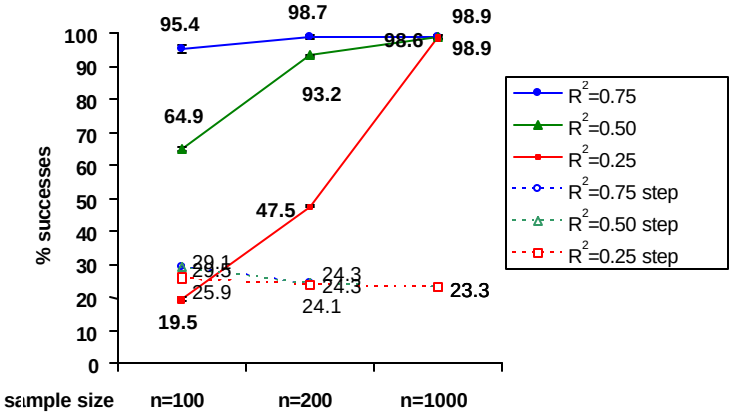
3.4 RETINA vs. stepwise regression, DGP6, 5% outliers.

In this experiment we use DGP6, which incorporates outliers. Using exactly the same data for RETINA and for stepwise regression, we find that RETINA outperforms stepwise regression for all combinations of sample size and R^2, except for n=100 and R^2=0.25. The percentage of successes of stepwise is decreasing with n and does not improve with R^2. Stepwise consistently overparameterizes between 58 and 76% of the time, adding on average between 2.44 and 3.12 extra regressors.

Table 7S. Dgp6, linear with 5% outliers in u, RETINA and stepwise

	n=100	n=200	n=1000
R^2=0.75	95.4	98.7	98.9
	0.66	0.36	0.33
	1.32	0.72	0.66
R^2=0.75 step	29.5	24.3	23.3
R^2=0.50	64.9	93.2	98.9
	1.51	0.8	0.33
	0.66	0.36	0.33
R^2=0.50 step	29.1	24.3	23.3
R^2=0.25	19.5	47.5	98.6
	1.25	1.58	0.37
	0.66	0.36	0.33
R^2=0.25 step	25.9	24.1	23.3

Figure 7S. Dgp4, linear 5% outliers in u, RETINA and



Average number of excess regressors in overparameterized models, stepwise in Table 7S Dgp4, 5% outliers, and % overparameterizations.

	n=100	n=200	n=1000
R^2=0.75	2.78	2.54	2.44
	70.50%	75.70%	76.70%
R^2=0.50	2.81	2.54	2.44
	69.40%	75.70%	2.44%
R^2=0.25	3.12	2.55	2.44
	58.40%	74.70%	76.70%