An Empirical Test of the Weighted Effect Approach to Generalized Prediction Using Recursive Neural Nets[1]

Rense Lange

University of Illinois at Springfield
Springfield, Il 62794-9243
lange@uis.edu

Abstract

The requirement of a strict and fixed distinction between dependent variables and independent variables, together with the presence of missing data, typically imposes considerable problems for most standard statistical prediction procedures. This paper describes a solution of these problems through the "weighted effect" approach in which recursive neural nets are used to learn how to compensate for any main and interaction effects attributable to missing data through the use of an "effect set" in addition to the data of actual cases. Extensive simulations of the approach based on an existing psychological data base showed high predictive validity, and a graceful degradation in performance with an increase in the number of unknown predictor variables. Moreover, the method proved amenable to the use of two-parameter logistic curves to arrive at a three way "low," "high," and "undecided" decision scheme with a-priori known error rates.

Introduction

Most texts on statistical methods (e.g., [1]) present the topic of prediction as the problem of finding an optimal function to predict a set of unknown (dependent) variables from a disjoint set of known (independent) variables. Although this approach is appropriate when testing hypotheses, it may not be applicable in many applied contexts. For instance, when diagnosing a client, physicians and psychologists typically have access to a personal file containing the outcomes on standard tests, together with diverse items of information already gathered by other professionals. Depending on the nature of the case, the distinction between dependent and independent variables is often blurred because: (a) some dependent variables may already be known, whereas some of their indicators (independent variables) are lacking; (b) the same conclusion can often be reached based on different sets of available information; and (c) at any point, practitioners have the option to gather additional information. Since the same variable may occur both as a dependent variable and as an independent variable, the resulting process defies description in terms of a rigid a-priori distinction between these two types of variables.

Situations such as the above would benefit greatly from knowledge discovery systems that allow any set of unknown variables to be predicted from arbitrary collections of already known variables. To achieve this goal in the traditional framework, one might propose to simply recompute the predictor function each time a new situation arises. Where feasible from a computational and statistical point of view, this approach would require constant access to all appropriate databases. Unfortunately, such databases are often confidential or proprietary, and hence this solution is rarely feasible for large scale applications. Alternatively, one might consider computing a different predictor function for each conceivable pattern of known and unknown variables. However, this approach soon breaks down from a computational point of view. For instance, in a relatively simple application with 30 variables there are a maximum of 2^{30} different patterns of known and unknown variables. Even if only 0.1% of these combinations did actually occur, this would still leave over 1 million cases to be considered.

The present research describes an approach to knowledge discovery that relies on the capability of recursive neural nets to store information such that their performance benefits from knowing which variables can or cannot be used in the prediction process. Because known variables are allowed to differ with respect to their contribution to a particular prediction, the resulting procedure is called a "weighted effect" approach. Following an outline of this approach, later sections describe an empirical evaluation based on an actual data set. Finally, to determine the practical potential of the approach, particular attention is paid to the validity of its predictions.

[1] The author wishes to thank James Houran and Joel Adkins for their comments on an earlier version of this paper I further acknowledge Wayne Penn and Naomi Lynn for their vivid demonstration of the urgent need for more sophisticated thinking tools in order to avoid erroneous conclusions derived by naïve analyses of quantitative data.

A Weighted Effects Approach

To facilitate the presentation, the following terminology is introduced:

(i) The set **V** (with elements *v*) contains all variables that are relevant in a particular context.

(ii) All *known* variables are contained in the subset **IN** $\subset$ **V**.

(iii) All unknown variables are contained in the subset **OUT** $\subset$ **V**.

(iv) For reasons that will become clear later, predictions are the result of the mapping:

$$\mathbf{V'} = \mathbf{F}(\mathbf{V},\mathbf{E}),$$

where **E** represents the "effect" of knowing [or *not* knowing] the value of each variable *v*. That is, *all* variables in **V** (including the unknown variables in **OUT**) are input to the function **F** to yield an new vector **V'** with predicted values (including predictions for already known variables in **IN**), as moderated by the effects in **E**.

The author proposes that the effect set **E** should contain an entry for every variable in **V**, such that $e_i = 1$ when the value of the variable is known, and $e_i = -1$ when its value is not known. The rationale behind this assumption is that such weights allow a neural net to learn all linear main and interaction effects [and perhaps some non-linear effects also] associated with the presence or absence of each variable. This claim follows from the observation that the **IN** vs. **OUT** status of the variables in **V** can be thought of as creating a *v*-factorial 2 x 2 x ... x 2 analysis of variance (ANOVA) design. It has long been known [2] that any ANOVA design can be translated into an equivalent multiple regression problem through the use of "coded dummy variables." In our particular case where each factor has only two levels, the situation simplifies considerably since all main and interaction effects can be captured by single parameters. Also, in this case the required number of orthogonal dummy variables is equal to number of main and interaction effects.

For instance, assume that we are interested in estimating a dependent variable *Y* from the two predictor variables *A* and *B* in a 2 x 2 ANOVA design. By defining the three sets of orthogonal dummy weights e_a, e_b, and e_{ab}, least- squares estimates of all main and interaction effects can be obtained by minimizing the expression:

$$\Sigma_i\Sigma_j\,(e_a.\alpha + e_b.\beta + e_{ab}.\chi + \mathrm{K} - y_{ij})^2,$$

where α represents the main effect of *A*, β represents the main effect of *B*, χ represents the *A x B* interaction, and K denotes an additive constant. The procedure is illustrated in Figure 1 which shows the predicted y_{ij} as the sum of the dummy variables e_a, e_b, and e_{ab} weighted by their corresponding effects α, β and χ.

e_a	e_b	e_{ab}	
$-1.\alpha$	$-1.\beta$	$-1.\chi$	$= y_{00}$
$-1.\alpha$	$+1.\beta$	$+1.\chi$	$= y_{01}$
$+1.\alpha$	$-1.\beta$	$+1.\chi$	$= y_{10}$
$+1.\alpha$	$+1.\beta$	$-1.\chi$	$= y_{11}$

Figure 1: *Prediction of Y based on the main effects α and β, and their interaction χ in a multiple regression framework using the dummy weights* e_a, e_b, and e_{ab}.

Note that for *n* predictor variables there are a total of $2^n - (n + 1)$ possible interaction terms, leading to a combinatorial explosion that contradicts our assumption that **E** should contain only *v* items. Notice however that the e_a, e_b, and e_{ab} columns in Figure 1 are related as if:

$$e_{ab} = e_a \text{ XOR } e_b,$$

provided that the weights +1 and -1 are replaced by the truth values T and F, respectively. Thus, the dummy variable e_{ab} need *not* be provided explicitly because it can be learned from e_a and e_b via backpropagation in neural nets with at least one intermediate layer.

In general, the weighted effect approach is predicated on the assumption that interaction effects need not be represented explicitly in **E** because they can be learned selectively from training data on an "as needed" basis by neural nets. Thus, the number of interactions that can be accommodated will depend on the size of the layers of the neural net. Also, the order of the interactions is limited by the number of intermediate layers. For example, analogous to XOR problems with three variables, three way interactions require at least two intermediate layers. Thus, the weighted effect approach is best used in situations where higher order interactions are rare. However, such interactions are perhaps best avoided anyway because they typically generalize poorly to new cases.

An Empirical Test

Naturally, the above considerations do not guarantee that adding any main and interaction effects due to (not) knowing predictor variables will improve prediction in actual practice. For this reason, the next sections describe the results of an empirical performance study based on an existing body of psychological data. In particular, all results reported below were derived from a database of 721 cases of hallucinatory episodes experienced by ostensibly normal individuals as described in detail in a psychological study published by Lange et al. [3]. Each case consists of 31 variables, 3 of these are considered continuous, but most (28) are binary categorizations. Since the present research focuses primarily on the predictive quality of the weighted effect approach, the following presentation addresses only the statistical properties of the results. Consequently, variables are simply referred to by their ordinal position in the original database (i.e., 1 through 31), and readers interested in content oriented issues are referred to the aforementioned paper.

Design. In order to test whether addition of the effect set **E** improves prediction, two basic experimental conditions were created:

> In the *Effect condition*, the effect set E is defined as described in the preceding section, **V** is a copy of a record in the data set, and the function **F** was implemented as a partially recursive neural net. This net has as its outputs the 31 variables in **V'** and as its inputs the 31 + 31 = 62 variables in **V** and **E**. A standard backpropagation algorithm [4] was used with logistic squashing functions over the range -1.0 to +1.0. All variables were scaled over the range -0.9 to +0.9. The 721 cases were randomly divided into a training set and a test set of approximately equal size (i.e., n = 360 or n = 361).
>
> *Training Phase.* During the training phase, knowledge of predictor variables (or the lack thereof) was simulated by randomly selecting between 0 and 15 variables, using an efficient algorithm described in [5, p. 122]. The thus selected variables were assigned the value 0.0, they were added to **OUT**, and their entry in **E** was set to -0.9. The remaining variables kept their original values, were added to **IN**, and their entry in **E** was set to +0.9. The resulting **V** + **E** were then presented to the backpropagation algorithm. Over 5000 iterations were used to train each of two types of nets: Effect [1], a partially recursive net with one intermediate layer, and Effect [2] a partially recursive net with two intermediate layers.
>
> *Test Phase.* The weights obtained during the training phase were validated on the test set by randomly selecting between 1 and 15 variables for inclusion in the **OUT** set, and the **E** set was constructed and used exactly as during the forward propagation stage in the training phase. Care was taken to insure that each variable occurred in the **OUT** set about 10,000 times.
>
> The *"Control" condition* was identical to the Effect condition with the exception that no set **E** was used during the training phase. Instead, randomly selected elements of **OUT** were simply assigned the value 0. That is, all results are based on a fully recursive neural net with 2 intermediate layers, 31 outputs (**V'**), but only 31 inputs (**V**). This net is referred to as Control [2].

Main Hypothesis. Figure 2 (solid lines) shows the coefficients of determination (i.e., Pearson r^2) between the actual and the predicted values of the variables in **V** as computed over the cases in the test set. The values shown are the average over all 31 variables. The crucial comparison is between the Effect and Control models with two intermediate layers. It can be seen that our basic hypothesis is supported because the average predictive validity is consistently greater for Effect [2] than for Control [2], for **OUT** sets ranging in size from 1 to 15 unknown variables.

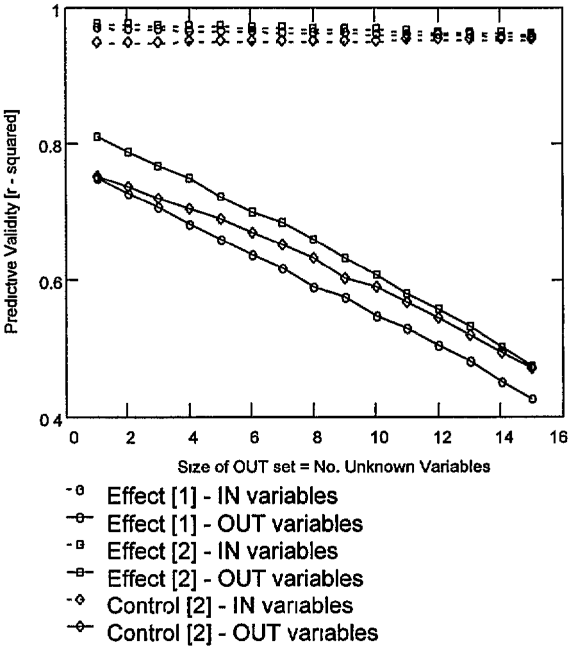

Figure 2: *Average Predictive Validity [Pearson r^2] Over All 31 Variables for Effect Models [1] and [2] and Control Model [1] by size of* ***OUT*** *set. (Test data only)*
NOTE: *The definitions of Effect [1], Effect [2], and Control [2] are given in the preceding Design section.*

The performance of Control [1] catches up for larger **OUT** sets. However, it behaves more like the simpler Effect [1] model for smaller **OUT** sets. Nearly identical findings were obtained when using RMS as a performance criterion. Thus, these results clearly show that the effect set **E** improved overall predictive validity.

Although only of tangential importance for the present purposes, Figure 2 also shows that the predictive validity of Control [2] is consistently lower for **IN** variables (diamonds + dotted lines). In other words, the addition of the effect set **E** to the inputs served to minimize distortions in *known* variables in the transformation from **V** to **V'**.

Response Curves

Of particular interest for present validation purposes is the likelihood that a person will be classified correctly as either *"low"* or *"high"* on some dependent variable of interest based on knowledge of his or her predicted score *v'*. Because most of our variables represent binary categorizations, a person was classified as *"high"* on a variable if this person's *actual* score exceeded 0.0, and the person was classified as *"low"* otherwise. As is customary, the resulting relation between the predicted and the actual score is assumed to follow a logistic "response" curve of the form:

$$P(\textit{"high"} \mid v') = \{1 + e^{-(p + q \cdot v')}\}^{-1},$$

where the parameters *p* and *q* can be estimated via standard maximum likelihood methods.

For instance, Figure 3 compares the performance of a particular variable (No. 28) under Effect [1] and Effect [2] as derived from about 10,000 data points generated during the training phase (for details see next section). Note that the response curve under Effect [1] (left panel) breaks down over the range $0.2 < v' < 0.5$. Adding an intermediate layer greatly improves this situation as it appears that the Effect [2] net (right panel) smoothes the response curve nicely while providing a "patch" for $0.2 < v' < 0.5$.

Three-Way Decisions. In the following, response curves were used to create a *three-way* decision scheme with arbitrary and adjustable error rates. This can be achieved by selecting two appropriate percentiles in the logistic curve. For instance, the right panel in Figure 3 shows the 40-th and 60-th percentiles (P_{40} and P_{60}) of the logistic curve. By classifying people as *"low"* on Variable 28 if their predicted score falls below P_{40} and as *"high"* if their predicted score exceeds P_{60}, a third category, *"undecided,"* is defined for predicted scores between P_{40} and P_{60}. Naturally, the choice of percentiles used to define these three categories will depend on the nature of the application. In our case, P_{40} and P_{60} were deemed appropriate for all variables and these values were used throughout.

To determine the viability of this approach, the following experiment was performed. *First*, analogous to the procedure described in the design section, the training data set and the connection weights from the training phase were used to generate predictions from randomly constructed **OUT** sets (and corresponding **E** sets) with between 0 and 15 unknown variables. The known classifications and their corresponding predicted values *v'* were used to generate logistic parameters *p* and *q* for all variables, using about 10,000 observations per variable. *Second*, the thus obtained parameters *p* and *q*, as well as the already existing connection weights, were then used to generate predictions over the test data set. In particular, randomly constructed **OUT** sets (and corresponding **E** sets) were used with between 1 and 15 unknown variables. Each prediction v' was then classified as "low," "undecided," or "high" depending on the values of P_{40} and P_{60}.

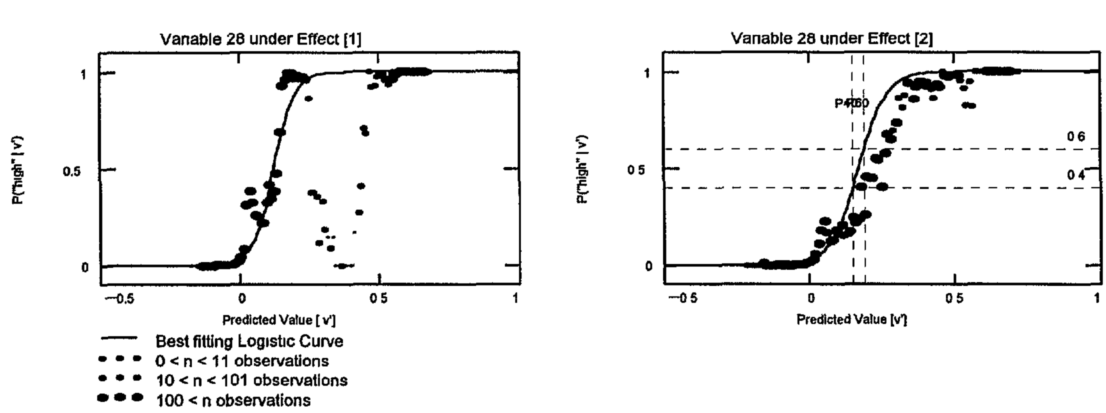

Figure 3: *Best Fitting Two Parameter Logistic Curve and Actual Observations for Variable 28 Under Effect [1] and Effect [2]. (NOTE: Training data only).*

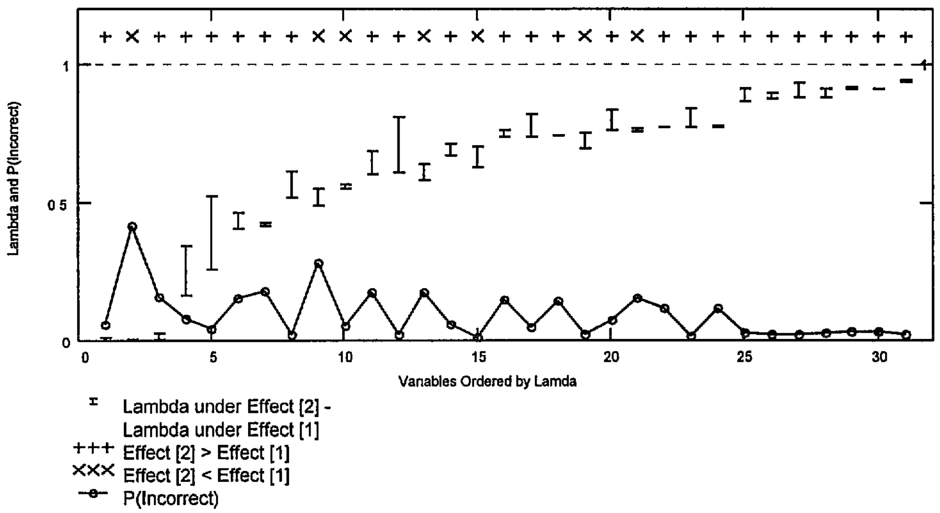

Figure 4: *Proportion of Erroneous Classifications and the Predictive Validity of All 31 Variables Ordered by Their λ Coefficients for Effect [1] and Effect [2]. (Test data).*

Because our main interest concerns the correctness of the classification as either "low," "undecided," or "high," predictive validity cannot be determined via Pearson correlation's or RMS values. Instead, the directional and non-parametric λ coefficient was used. The value of this coefficient can be interpreted directly as the proportional increase in classification correctness as the result of knowing the predicted value v'. Figure 4 shows the λ values for all 31 variables under Effect [1] and Effect [2] averaged over the 15 **OUT** conditions and ordered by the combined λ magnitudes. It can be seen that the predictability increased by more than 50% for 25 out of the 31 variables, but that the leftmost few variables performed very poorly. It should be pointed out, however, that the low λ values occur mainly for variables showing little variation. Because such variables can be predicted reliably from their modal values, the overall error rate remains relatively low (*M = 0.091*). In fact, the solid line in Figure 4 indicates that only one variable (the second worst) has an overall error rate in excess of 0.3.

Highly similar conclusions were obtained from an analysis of the error rates for the individual categorization as either "low," or "high." Figure 5 shows the average probability of being classified as "high" given that v' > P_{60} (solid curves), or that v' < P_{40} (dotted curves). It can be seen that all but the first three variables perform very satisfactory.

A Comparison of Effect [1] vs. Effect [2]

Throughout the preceding, Effect [2] was slightly superior to Effect [1] in various respects. For instance, Figure 4 (top row) indicates that Effect [2] yielded higher λ values for 24 of the 31 variables (*M = 0.655 vs. M = 0.617*). Also, Effect [2] resulted in slightly more correct overall classifications (*M = 0.913*) than Effect [1] (*M = 0.904*). Although the overall differences are very small, the addition of an intermediate layer had important consequences for some individual variables, as indicated by a detailed analysis of the relation between λ and the size of **OUT** for all 31 variables. Due to space limitations, Figure 6 shows this relation only for the "worst" performing variable (leftmost in Figures 4 and 5), the "best" performing variable (rightmost in Figures 4 and 5), and one "intermediate" performing variable (the fourth in Figures 4 and 5). It can be seen that the "best" variable does not benefit from the additional intermediate layer,

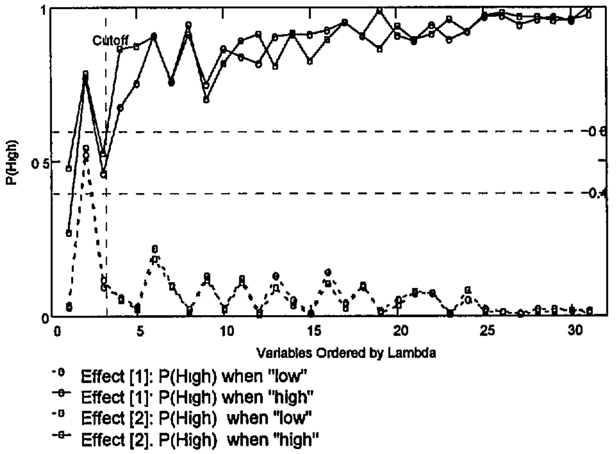

Figure 5: *Proportion "High" Classifications for Actually "High" or "Low" Cases Averaged Over **OUT** Sets of Size 0 to 15. (Test data).*

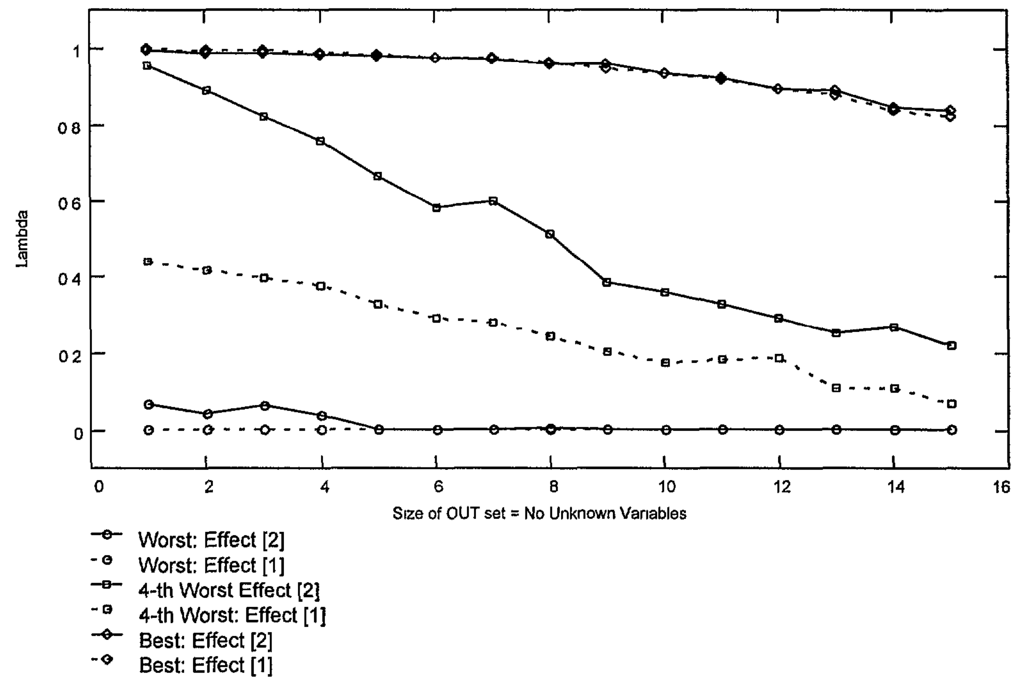

Figure 6: *Predictive Validity λ of Three Variables as a Function of the Size of **OUT** for Effect [1] and Effect [2] (Test data).*

and that the improvement for the worst variable is limited to small **OUT** sets. However, the "intermediate" variable shows dramatic improvements over the entire range. It is recommended therefore that the decision concerning the desired number of intermediate layers should take the performance of individual variables into account, and that it should not solely be based on overall performance indices.

Discussion

The results presented in this paper strongly support the weighted effects approach since it proved possible to find a single framework to accommodate prediction in situations with varying numbers of dependent and independent variables. In addition, the case study indicated that the method is sufficiently flexible and powerful for practical application, and the current results have already led to new research in psychology and criminology. Further, given that all information is contained in the net's connection weights, it is possible to provide predictions *without* having to provide access to the original data set. For this reason, research is currently underway to provide an expert system type user interface similar to MACIE [4].

Additional study of the weighted effect approach seems desirable. For instance, it is not clear whether the present training approach is optimal or whether more efficient "training schedules" exist. Also, currently the net is used for *all* learning that takes place. It seems desirable, however, to be able to separate the learning associated with the effect set E from that associated with **V** in applications containing higher order interaction effects.

Finally, the author feels that further development might benefit from more detailed analyses of the statistical rationale underlying the weighted effect approach as this might lead to increased control over the quality of prediction.

References

[1] Scott, E.M., and Delaney, H.D. (1990). *Designing experiments and analyzing data.* Belmont, CA: Wadsworth.
[2] Cohen, J. (1968). Multiple regression as a general data-analytic system. *Psychological Bulletin*, 70, 426 - 433.
[3] Lange, R., Houran, J., Harte, T., and Havens, R. (1996). Contextual mediation of perceptions of hauntings and poltergeist-like experiences. *Perceptual and Motor Skills*, 755-762.
[4] Gallant, S.I. (1994). *Neural network learning.* Cambridge, Mass.: MIT Press.
[5] Knuth, D.E. (1969). *The art of computer programming.* Vol. II. Reading, Mass.: Addison-Wesley.