A LEARNING SYSTEM WHICH ACCOMMODATES FEATURE INTERACTIONS

Larry A. Rendell

Department of Computing and Information Science
University of Guelph
Guelph, Ontario, NIG 2WI Canada

ABSTRACT

The author's state-space learning system has effectively optimized the coefficients of linear evaluation functions. The incremental approach uses statistical performance measures from completed solutions to bootstrap the heuristic, which estimates probability of task usefulness. These statistics are clustered in feature space, forming a mediating knowledge structure (region set) between the direct performance measures and the generalized evaluation function. The regions are data-determined, insensitive to noise, and allow management of interacting features through natural piecewise linearity. Early experiment with nonlinearity indicates stability, flexibility and improved task performance.

1. INTRODUCTION

The evaluation function has frequently been used as a heuristic in what is called best-first search [1,2,4,93. A standard technique is to combine several more elementary functions or features. As argued in [3], forming a heuristic function from a set of features is theoretically as powerful as any other design. The problem though is to merge features usefully. Often the linear combination H = b.f is imposed, where b is the coefficient vector for the feature vector f, though this is generally insufficient [1,9]. Even with this restrictive formulation, b is difficult to optimize.

The optimization should be governed by some performance measure (such as number of states generated to reach the goal) but often no solution whatever can be found within resource constraints. Despite this impediment, some approaches have been very effective, e.g. [9]. In [8] the present author described a successful new basis for learning. The system implemented was able not only to solve the fifteen puzzle, but also to optimize feature coefficients for linear evaluation functions, a unique result. Since the scheme has good conceptual and experimental support, work is underway to improve it. One extension is to increase accuracy [7]; another is to accommodate feature interactions, to allow more general evaluation. The latter uses a natural piecewise linear method outlined in [6], and developed, implemented and tested here.

2. KNOWLEDGE STRUCTURE

Like other recent approaches [10], a penetrance learning system (PLS) uses completed searches of training problems. Unlike them, it computes statistics measuring solution density in feature space (Fig. 1). Although it is data driven, PLS is insensitive to noise since it is stochastic. The raw statistics, which depend on the problem instance set P attempted and heuristic H guiding the search, are called elementary penetrances p(r, H, P), where r is a (rectangular) feature space volume. From these data, a normalized true penetrance estimate p(r) is computed. This value is the estimated probability of a state A being in a breadth first solution of a random problem instance, given that A maps into r. Derived from repeated observations and incremental computations, the evolving evaluation function H is designed to predict true penetrance.

To house the true penetrance estimate p of a feature space volume r, a region R is defined to be the quintuple (r, c, *P,* e, b) . The second element c is the centroid, a representative of r. The final two elements relate to p: e is the error, an inverse measure of the reliability of p, and b is a coefficient vector, explained later. A set of these regions, the cumulative region set C, is both the control structure used by the problem solver and the knowledge structure improved by the learning element (Fig. 2). This set accumulatates information over several iterations, as its regions are incrementally resolved into smaller units just adequate to express known relationships. The result is an effective economy, a refinement of Samuel's [9] signature tables which did not alter data categories automatically.

3. LEARNING ELEMENT MECHANISMS

While [8] gives considerable detail of the original implementation PLS1, the main notions are summarized here to motivate further development. The incremental three step operation is pictured in Fig. 2. First, given some training problems and evaluation function, the solver extracts penetrance statistics from the resulting search trees. Secondly, the clusterer modifies the cumulative region set based on these penetrance measures. Finally, the new cumulative region set becomes data for the regresser, a curve fitting algorithm which generates an improved heuristic for the next iteration. Together the clusterer and regresser form the learning element.

The clusterer is complex. New regions are formed when penetrance data are found to diverge within any existing region R. This region refinement is realized by an efficient algorithm that repeatedly splits rectangles, until further differentiation is not warranted by the recent data. In any iteration after the first, these elementary penetrances are heavily biased by the heuristic used to obtain them, so a fine normalization procedure unbiases values within R. Thus, penetrances for newly split rectangles become commensurate with the true penetrance estimates of the cumulative regions. Region refinement is repeated in every iteration, so that knowledge is increasingly resolved.

A separate coarse normalization algorithm operating over the whole feature space obtains fresh true penetrance estimates for established regions. This algorithm assumes a pattern in the biased data. It smooths the true penetrance estimates of all established regions against the new elementary values within matching feature space rectangles to find a conversion factor to apply to all elementary penetrances. After normalization, these new values are averaged with the old to improve estimates. In this penetrance revision, weights for the averaging depend on the accuracy of each datum. The end result is decreased error in the combination, so that cumulative true penetrances gradually become more inert (although region refinement counters this trend).

All this manipulation by the clusterer is designed to provide proper data to fit true penetrance as a function of features for the solver's heuristic. Each region $R = (r, \underline{c}, \hat{p}, e, \underline{b})$ in the cumulative set C has an undefined feature coefficient vector $\underline{b}$ after clustering, but the regresser (Fig. 2) determines $\underline{b}$ from C. (The contribution of each R is inversely related to its error e.) The regresser rejects any useless or less general features by zeroing their coefficients, and decides the relative importance of the discriminating features by setting their coefficients to values best suiting C. The next section will describe how $\underline{b}$ now becomes a property of R, to accommodate feature interactions.

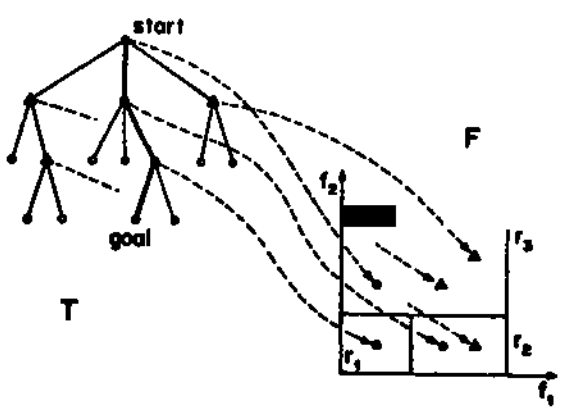

Fig. 1. Localized penetrance discriminates. Developed nodes from search tree T are mapped into feature space F. The whole space penetrance of T is 3/6, whereas localization in F gives (e.g.) three elementary penetrance values: $p(r_1,T) = 1/1$, $p(r_2,T) = 1/2$, and $p(r_3,T) = 1/3$.

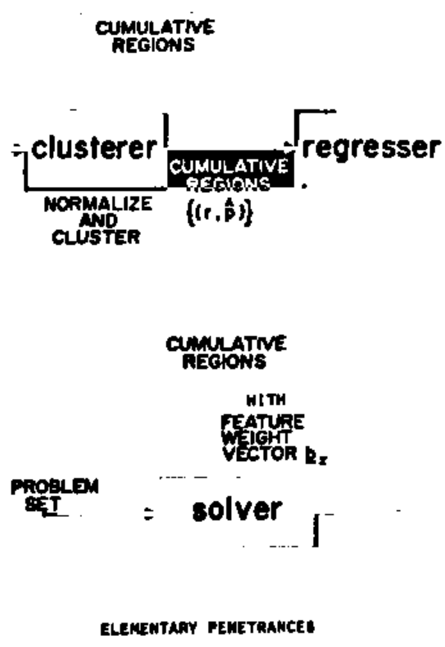

Fig. 2. Penetrance learning system PLS1. The essence of PLS knowledge is a set of feature space penetrance regions, used as the solver's heuristic and to accumulate experience.

4. PIECEWISE LINEARITY

This method takes advantage of the natural partitioning of the feature space into regions (c.f. [1, p.430], [5, p.317]) and allows increasing departure from linearity as the refinement improves in later iterations (as the number of regions increases). In this extension of PLS1, the clusterer remains unchanged while the regresser is altered.

Instead of a single penetrance-feature surface fitted over the whole space, there are now as many of these hyperplanes as regions in the cumulative set C. Each region R = (r, c, p, e, b_r) of C is viewed as the principal one for its own regression; the coefficient vector b_r is computed using an R-centered weighting of every contributing region $Q \in C$. As mentioned in the previous section, the regression is already weighted according to penetrance error e. In this new, piecewise linear design PLS1a, the former weight is multiplied by an additional factor related to the distance between R and Q, so that Q plays a greater role if it is near R. In the determination of this distance, the feature space is deformed to capture the relative importance of the various features. (Details of this and related aspects are provided in the appendix.)

Since each regression is still linear, the process is quite stable. (In contrast, permitting feature interaction by using higher order models requires many more coefficients and this 'uses up[1] the data.) At the same time the piecewise linear scheme PLS1a is flexible, allowing a continuously variable amount of nonlinearity in order to suit the current power of the entire learning system. This variability is mechanized by introducting a system parameter called the localization power L > 0 as an exponent for the distance measure (again, refer to the appendix).

To test the utility of this more sophisticated learning element, the solver was altered so that two new modes of evaluation can be selected. The first, discrete piecewise linear procedure simply predicts the true penetrance of a state A according to the local heuristic function of the region into which A maps. The more complex evaluation mode, smooth piecewise linear, uses all regions in the cumulative set In every evaluation, employing a distance weighting like the one above.

Prelimarary program runs have been made to discover characteristics of the scheme: its utility, cost, and stability. The cumulative region set used was a four dimensional one for the fifteen puzzle [8]. Since these four features were originally used in the strictly linear system, they had been deliberately selected for low interaction. Hence this is a mild test of PLS1a. In the solver, the cost increase of discrete piecewise linear evaluation over the strictly linear PLS1 mode is negligible, but edge effects (discontinuities) vitiate the scheme; performance is very poor. On the other hand, smooth piecewise linearity seems promising; its cost is also low. Results are shown in Fig. 3, where the extent of nonlinearity is varied by choice of the localization power L. In this curve the optimum is attributed to two conflicting factors: As L is increased some advantage occurs because the relationships are inherently nonlinear, and nearby regions now play a justifiably bigger part in the determination of each local heuristic. However, distant regions, which formerly had a stabilizing role, now have a diminished influence, so there is a general loss of support. The inaccuracy and graininess of individual regions gradually overpower the benefit of localization.

An important property of PLS1a is its stability. When PLS1 was used with higher order models instead of piecewise linearity, performance was degraded. Also in contrast, PLS1a allows easy observation of the relative importance of features in any area of the space, since simple feature weighting is used. Furthermore, relationships exemplified by Fig. 3 are useful. The magnitude of the optimal localization power is a measure of region accuracy, and indirectly, of the utility of the entire learning system.

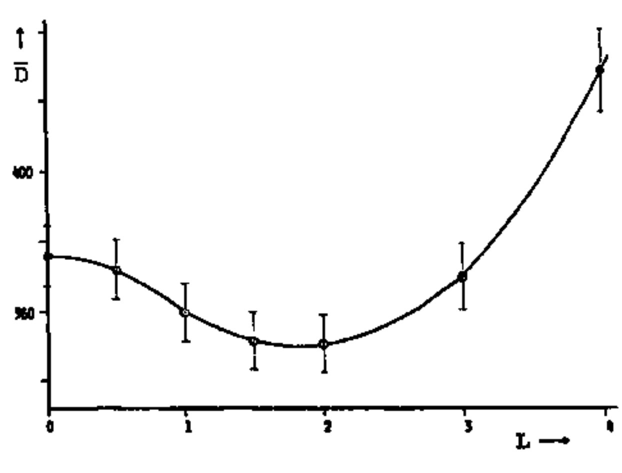

Fig. 3* Variation of performance with degree of nonllnearity. Shown is average number of nodes developed D before solution In a random sample of 1000 puzzles, vs. localization power L. 95% oonfldenoe intervals are indicated.

5. CONCLUSIONS

The piecewise linear scheme PLS1a is natural, flexible and stable. Its low cost and performance improvement warrant further investigation. The next step is to attempt stronger feature interaction, with support from a scheme designed to improve the accuracy of true penetrance estimates [7]. The freedom to vary the localization power L will facilitate experimentation in determining the general utility of PLS1a as a heuristic learning system.

REFERENCES

1. Berliner, H., Experiences in evaluation with BKG — a program that plays backgammon, Proc. IJCAI-77, 428-433.
2. Doran, J. and Michie, D., Experiments with the graph-traverser program, Proc. Roy. Soc, A,_ 294 (1966)
3. Feldman, J.A. and Yakimovsky, Y., Decision theory and artificial intelligence: I. A semantics-based region analizer, Artificial Intelligence 5 (1974) 349-371:
4. Gaschnig, J.G., Performance measurement and analysis of certain search algorithms, Dept. of Computer Science CS-79-124 (PhD thesis), Carnegie-Mellon University, 1979.
5. Michie, D. and Ross, R., Experiments with the adaptive graph-traverser, in Meltzer, B. and Michie, D. (eds.), Machine Intelligence 1, (1970) 301-318.
6. Rendell, L.A., State-space learning systems using regionalized penetrance, Proc. CSCSI/SCEIO-82, 150-157.
7. Rendell, L.A., A doubly layered, genetic penetrance learning system, C.I.S. Department Report CIS82-16, University of Guelph, 1982.
8. Rendell, L.A., A new basis for state-space learning systems and a successful implementation, Artificial Intelligence £0, July 1983.
9. Samuel, A.L., Some studies in machine learning using the game of checkers II—recent progress, IBM J. Res, and Develop. 11 (1967)~Tu"1-5T7.
10. Sleeman, D., Langley, P., t Mitchell, T.M., Learning from solution paths: An approach to the credit assignment problem, AI Magazine 3, (1982) 48-52.

APPENDIX - LOCALIZATION AND DISTANCE

While section 4 gave a general picture of the piecewise linear method, this appendix details the localization. Let the principal region from a cumulative set C be R = (r, c, p, e, b_r). Recall from section 3 that the true penetrance estimate p and the feature space centroid c are used along with others from C in a regression to determine b_r for the true penetrance predictor H_r = exp b_r.f. Here H_r is regionalized; the weight for each region Q c C contributing to the R-centered regression is to depend on the distance of Q from R. Before constructing this distance measure, we need to consider that features are not uniformly important; in fact a feature can be completely irrelevant. Hence the distance, itself, is weighted by b_r. This of course is circular since it is b_r which is to be determined. However, the procedure is iterative: First the global coefficient vector b is calculated, weighting each region equally, then this estimate of b_r is used, repeating until the value converges. Even this doubly multiple regression costs little compared with the time required by the solver.

The exact weighting is as follows: The distance factor is 1 / dist(R, Q). Expressing the ith estimate of $\underline{b}_r$ as $\underline{b}_r^{(i)}$, the function dist(R, Q) = $|\underline{b}_r^{(i)} \cdot (\underline{c}_r - \underline{c}_q)|^L$, if R ≠ Q; and $\bar{d}_r$, if R = Q. In this, $\underline{c}_r$ and $\underline{c}_q$ are the centroids of R and Q, and $\bar{d}_r$ is the average value of $|\underline{b}_r^{(i)} \cdot (\underline{c}_r - \underline{x})|^L$, over all points $\underline{x}$ within r. The exponent L ≥ 0, the localization power, decides the degree of non-linearity.

The solver uses one of two evaluation methods. The first, discrete piecewise linear procedure simply predicts the true penetrance of a state A to be H(A) = exp [$\underline{b}_r \cdot \underline{f}$(A)] where $\underline{f}$ is the feature vector and f(A) ∈ r. The more complex evaluation mode, smooth piecewise linear, uses all regions Q in the cumulative set C, each one weighted according to its distance from $\underline{f}$(A). If d_j = dist ($\underline{f}$(A), Q_j) for each of the J regions Q_j of C (1 < j < J), with the localization power L here fixed at 2, and if the coefficient vector of Q_j is $\underline{b}_j$, the predicted true penetrance of A is H(A) =

$$\exp\left(\sum_{j=1}^{J} [\underline{b}_j \cdot \underline{f}(A) / d_j] \Big/ \sum_{j=1}^{J} [1 / d_j] \right).$$