Improving an Adaptive Image Interpretation System by Leveraging

Lihong Li   Vadim Bulitko   Russell Greiner   Ilya Levner
University of Alberta
Department of Computing Science
Edmonton, Alberta, Canada T6G 2E8
{lihong|bulitko|greiner|ilya}@cs.ualberta.ca

Abstract

*Automated image interpretation is an important task in numerous applications ranging from security systems to natural resource inventorization based on remote-sensing. Recently, a second generation adaptive machine-learned image interpretation system (ADORE) has shown expert-level performance in several challenging domains. Its extension, MR ADORE, aims at removing the last vestiges of human intervention still present in the original design of ADORE. Both systems treat the image interpretation process as a finite-horizon non-discounted Markov decision process guided by a machine-learned heuristic value function. Therefore, the key task in automated development of such systems lies with acquiring the optimal control policy. This paper employs a new leveraging algorithm for regression (*RESLEV*) to improve the learnability of the optimal value function in MR ADORE. Experiments show that* RESLEV *improves the system's performance if the base learners are weak. More surprisingly, empirical evidence indicates that reducing the regression error can lead to worsening the overall performance of the system. We discuss this phenomenon thereby opening an exciting novel research direction.*

Keywords: adaptive image interpretation system, boosting, leveraging for value function regression, sequential decision making, reinforcement learning.

1. Introduction

Image interpretation is an important and highly challenging problem with numerous practical applications. Hand-crafted image interpretation systems suffer from an expensive design cycle, a high demand for human expertise in both subject matter and computer vision, and the difficulties with portability and maintenance. Over the last three decades, various *automated* ways of constructing image interpretation systems have been explored [7].

One of the promising approaches to automatic acquisition of image interpretation systems lies with treating computer vision as a control problem over a space of image processing operators. Early attempts used the schema theory [2, 3]. While presenting a systemic way of designing image interpretation systems, the approach was still *ad-hoc* in its nature and required extensive manual design efforts [9].

In the 1990's the second generation of control policy based image interpretation systems came into existence. More than a systematic design methodology, such systems used theoretically well-founded machine learning frameworks for automatic acquisition of *control strategies* over a space of image processing operators. The two well-known representatives are a Bayes net system [21] and a Markov decision process (MDP) based system [8].

The latter system ADORE (ADaptive Object REcognition) [8] and its extension MR ADORE (Multi-Resolution ADaptive Object REcognition) [17] learn dynamic image interpretation strategies for finding target objects in images. As with many vision systems, they identify objects in a multi-step process. The input is a raw image, and the output is an interpretation identifying target objects in the image; in between, the data can be represented as intensity images, probability images, edges, lines, or curves. The systems model image interpretation process as a Markov decision process, where the intermediate representations are continuous state spaces, and the vision procedures are actions. The goal is to learn a dynamic control policy that selects the next action (i.e., image processing operator) at each step so as to maximize the quality of the final interpretation. Instead of learning the policy directly, the system *learns* the optimal value function as the heuristic for the MDP-based policy.

In this paper we consider the problem of ensemble learning (in particular, leveraging [20]) for the value function in MR ADORE. The task of recognizing tree canopies from aerial photographs (i.e., labeling pixels belonging to tree canopies in an input image) is used as the testbed.

The rest of the paper is organized as follows. Section 2 reviews the requirements and design of MR ADORE. Sec-

(a) Input image (b) Desired output

Figure 1. The left image is an original photograph. The right one is the corresponding desired labeling provided by an expert as a part of the training set.

tion 3 gives a brief overview of boosting/leveraging methods. Section 4 presents and discusses the experimental results. Finally, section 5 concludes the paper and points out several interesting future research directions.

2. MR ADORE design

MR ADORE was designed with the following objectives: (i) rapid system development for a wide class of image interpretation domains; (ii) low demands for subject matter, computer vision, and AI expertise on the part of the developers; (iii) accelerated domain portability, system upgrades, and maintenance; (iv) adaptive image interpretation wherein the system adjusts its operation dynamically to a given image; (v) user-controlled trade-offs between recognition accuracy and resources utilized (e.g., time required).

2.1. Overview

The objectives above favor the use of readily available off-the-shelf image processing operator libraries (IPL). However, the domain independence of such libraries requires an *intelligent policy* to control the application of library operators. Operation of such a control policy is a complex and adaptive process [5]. It is *complex* in the sense that there is rarely a one-step mapping from input images to their interpretations; instead, a series of operator applications are required to bridge the gap between raw pixels and semantic objects. Examples of the operators include region segmentation, texture filters, and the construction of 3D depth maps (see Figure 2 for a simplified example).

Image interpretation is an *adaptive* process in that there is no fixed sequence of actions that will work well for most images. For instance, the steps required to locate and identify isolated trees are different from the steps required to find connected stands of trees. Figure 3 demonstrates two specific forestry images that require significantly different operator sequences for satisfactory interpretation results.

The success of MR ADORE therefore depends on its control policy: given an input image, how to select a sequence of operators to interpret the image most effectively?

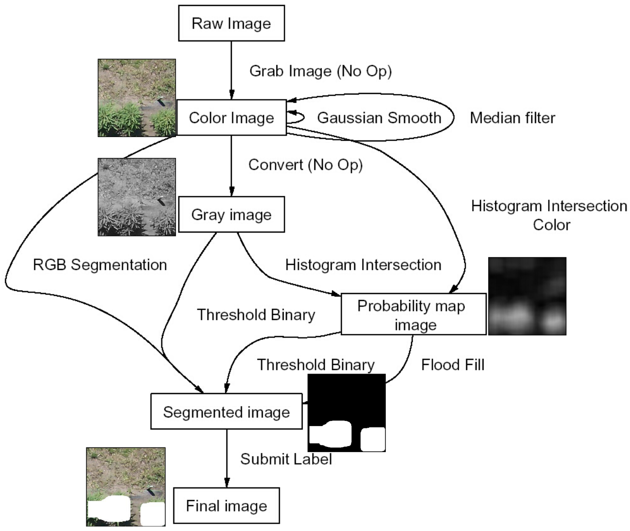


Figure 2. Data types (the boxes) and image processing operators (the arcs) in MR ADORE. Representatives of data tokens of each types are shown next to the nodes.

2.2. Learning control policies

MR ADORE starts with the Markov decision process (MDP) as the basic mathematical model by casting the IPL operators as the MDP actions and the results of their application as the MDP states. Then it learns an approximation to the optimal value function.

First, it uses training data (here, annotated images) to provide relevant domain information. Each training datum is a source image, annotated by an expert with the desired output. Figure 1 demonstrates a training datum in the forestry image interpretation domain.

Second, during the off-line stage the state space is explored via limited depth expansions of training images. Within a single expansion, all sequences of IPL operators up to a certain user-controlled length are applied to the training image. Since training images are user-annotated with the desired output, terminal rewards can be computed based on the difference between the produced labeling and the desired labeling. Then, dynamic programming methods [4] are used to compute the optimal value function for the explored parts of the state space. Note that MR ADORE does not use a discounting factor, making the entire problem a finite horizon non-discounted MDP. We represent the value function as $Q : S \times A \to \mathbb{R}$ where S is the set of states (image tokens) and A is the set of actions (IPL operators). The optimal $Q(s, a)$ computes the maximum cumulative reward the optimal policy can expect to collect by taking action a in state s and acting optimally thereafter [24]. As the raw state descriptions are on the order of mega-bytes each, we first abstract each state s into a set of features $f(s)$ using an abstraction function $f(\cdot)$. Then supervised ma-

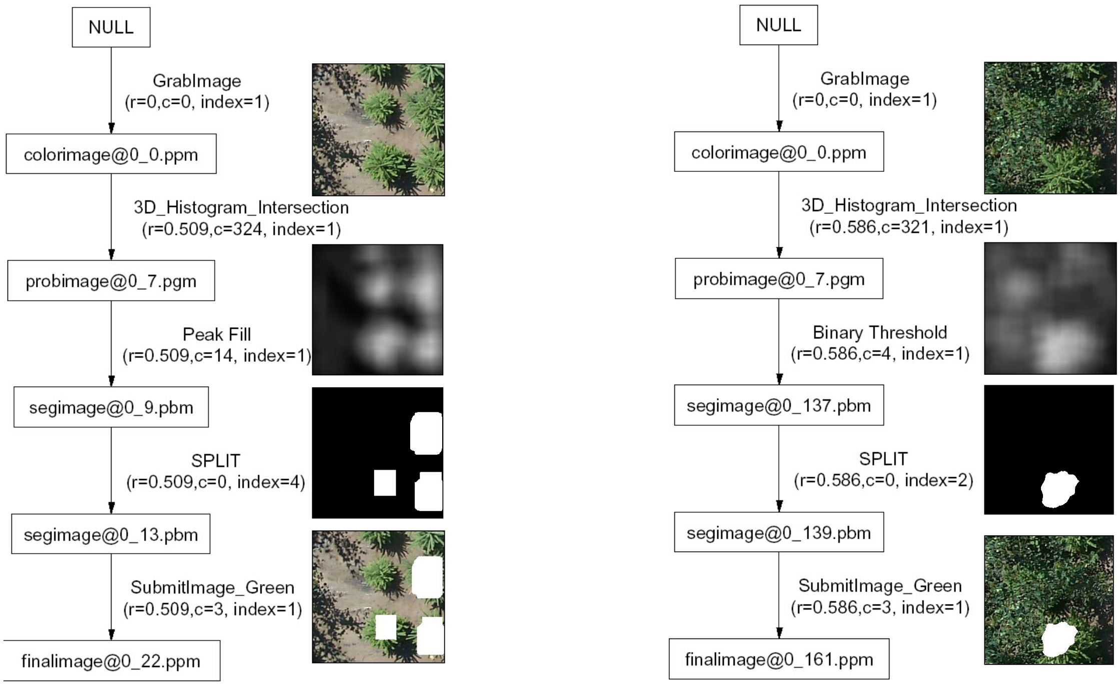


Figure 3. Adaptive nature of image recognition: two different input images require significantly different satisfactory operator sequences. Each node is labeled with its data type. Each arc between two data tokens is shown with the operator used.

chine learning extrapolates the sample Q^*-values computed by dynamic programming on the explored fraction of the state space onto the entire space.

Finally, when presented with a novel input image to interpret, MR ADORE first computes the abstracted version $f(s)$, then applies the machine-learned approximation to the value function $Q(\cdot,\cdot)$ to compute $Q(f(s),a)$ for each IPL operator a; it then performs the action $a^* = \arg_a \max Q(f(s),a)$. The process terminates when the policy executes action `submit(⟨labeling⟩)` where `⟨labeling⟩` becomes the system's output.

3. Boosting and leveraging methods

Boosting and leveraging have their roots in the PAC (Probably Approximately Correct) [25] learning model. A learner L_s for concept class $\mathcal{C}$ is a (*strong*) PAC learner, iff with an *arbitrarily high* probability, L_s produces a hypothesis h with *arbitrarily high* accuracy. The requirement of being arbitrarily accurate with an arbitrarily high probability is removed for weak learners. For example, a learner that produces hypotheses a little better than random guessing is a weak learner. An algorithm has the *PAC-boosting property* iff it can boost any weak PAC learner to a strong PAC learner [11].

Both boosting and leveraging methods work by repeatedly producing base hypotheses with modified training data, then using them to extend and modify the training data set, and then combining them into a final hypothesis that are better than any individual base hypothesis. While boosting methods provably have the PAC-boosting property, leveraging methods may or may not.

Since Schapire proposed the first provable polynomial time boosting algorithm [22], a number of boosting and leveraging algorithms have been developed [6, 10, 12, 13, 14, 16]. They are conceptually gradient-descent algorithms that iteratively increase the accuracy of the final hypothesis by decreasing some associated *potential functions*. Recently a simple and straightforward leveraging algorithm called ResLev (Figure 4) [19] has demonstrated the ability to improve the accuracy of a hypothesis by iteratively learning the previous ensemble hypothesis' residuals (i.e., errors) of predictions on training data. Experiments on the Friedman datasets [15] indicate the algorithm's effectiveness [18, 19].

Definitions: Given a hypothesis h and a set of training data

$$S = \{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$$

the *sample error* of h on S is defined as:

$$\widehat{err}_S(h) = \frac{1}{N}\sum_{i=1}^{N}[y_i - h(x_i)]^2$$

Let $\mathcal{P}$ be a probability distribution on the whole instance space, the *generalization error* of h with respect to $\mathcal{P}$ is defined as:

$$err_{\mathcal{P}}(h) = \mathbf{E}_{\mathcal{P}}[y - h(x)]^2$$

RESLEV (S, R, T_{max})

Input: A set of training samples
$S = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$;
A base regressor R;
Maximum number of iterations T_{max}

1. Initialize ensemble hypothesis $H_0 \equiv 0$
2. For $t = 1, \cdots, T_{max}$ Do
   (a) Apply R on S and produce a base hypothesis h_t
   (b) $H_t = H_{t-1} + h_t$
   (c) Modify data in S: $\forall i, y_i = y_i - h_t(x_i)$
3. Output final ensemble hypothesis $H_{T_{max}}$

Figure 4. RESLEV: Residual Leveraging

where (x, y) is drawn randomly according to $\mathcal{P}$.

The following theorem gives a *sufficient* condition under which RESLEV will decrease the sample error, based on the relation between the training data and the sample error.

Theorem 1 [19] *Given any training set* $T = \{(u_1, v_1), (u_2, v_2), \cdots, (u_M, v_M)\}$*, if the learner* L *produces a hypothesis* h *that satisfies*

$$\sum_{i=1}^{M} v_i^2 \quad > \quad \sum_{i=1}^{M} (v_i - h(u_i))^2 = M \cdot \widehat{err}_T(h) \qquad (1)$$

then RESLEV *reduces the sample error with the learner* L.

RESLEV reduces sample error by repeatedly reducing the residuals of predictions over the training data. Assuming that the training data and test data are both drawn from an unknown distribution $\mathcal{P}$, it is expected that a small sample error implies a small generalization error [1], provided that the ensemble hypothesis is sufficiently simple [1].

4. Empirical evaluation

We applied RESLEV to learning the optimal value function $Q(\cdot, \cdot)$ in MR ADORE. Multi-layer feed-forward neural networks were used as the base learners/regressors. Common sets of features including RGB-HISTOGRAM, HSV-HISTOGRAM, HSV-MEAN, textural features, etc. were used. Experiments were run with combinations of different features and neural network topologies, as shown in Figure 5 and 6. Three measures: sample and generalization errors in the Q-function and the value of the resulting MR ADORE control policy were computed.[2]

Thirty two forestry aerial images with user-annotated labeling were used in our experiments. Since the training data are quite limited, leave-one-out cross-validation was employed for evaluation. In each run, one image was selected for testing while the other thirty one images were used to train MR ADORE (i.e., to learn the function $Q(\cdot, \cdot)$). The three performance measures were then averaged over all the thirty two runs.

Figure 5 shows the experimental results [3]. For comparison, experiments with the random policy (i.e., the system randomly selects an operator to apply on the current token) were tried and an average relative reward of 26.3% was attained. Obviously, learning in all experiments increases the system's performance: the average relative reward ranges from 53% to 85%.

Several observations are in order. First, the leveraging process reduces the regression errors in the Q-function when the base learners are weak (e.g., SET#4). On the other hand, it can actually *increase* the errors when the base learners are strong to begin with (e.g., SET#3 and #5). An investigation of this phenomenon revealed that with stronger base learners condition (1) of Theorem 1 is often violated.

Second, leveraging can often improve the value of the policy while *decreasing* the regression accuracy (e.g., with SET#2). The reverse phenomena takes place as well. Furthermore, such a divergence between the effects of leveraging on regression errors and the performance of the control policy was recorded even with boosting algorithms such as SQUARELEV.R [12] [18]. These observations provide empirical support for the following claim:

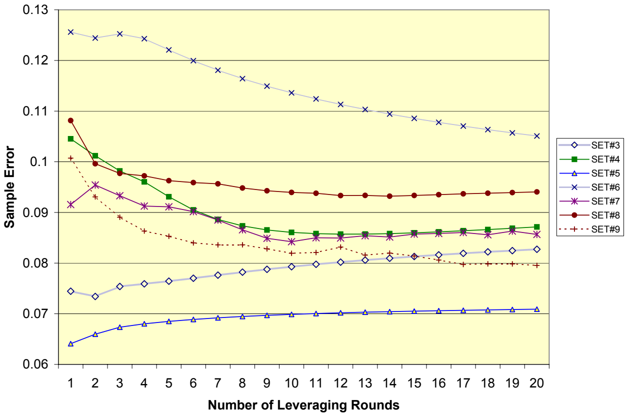
Claim: *decision-making problems behave differently from regression problems in terms of boosting/leveraging methods applied to the value function.*

Note that if the optimal value function can be learned with an arbitrary precision then the resulting control policy can be made arbitrarily close to the optimal policy (cf. [23] for an analysis of discounted cases). In practice, however, the complexity of sequential decision-making frequently does not allow to learn the optimal value function arbitrarily well. This paper demonstrate that in such cases and in the absence of discounting leveraging/boosting methods can have opposite effects on regression error vs. value of the resulting control policy. Therefore, it is of interest to develop leveraging and boosting methods that optimize the control policy value *directly*.

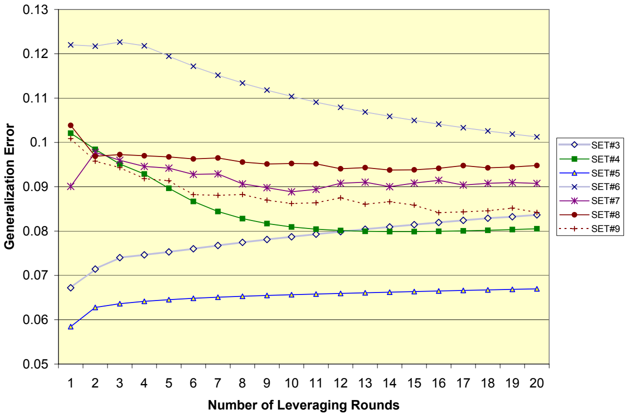
5. Conclusions and future directions

Future Work: As discussed in previous sections, primary future directions include: (i) employing other base regression algorithms so that the condition (1) in Theorem 1 is satisfied and thus RESLEV can be of help; (ii) investigation of the generalization error of RESLEV, and whether it has the PAC-boosting property. (iii) comparing the ef-

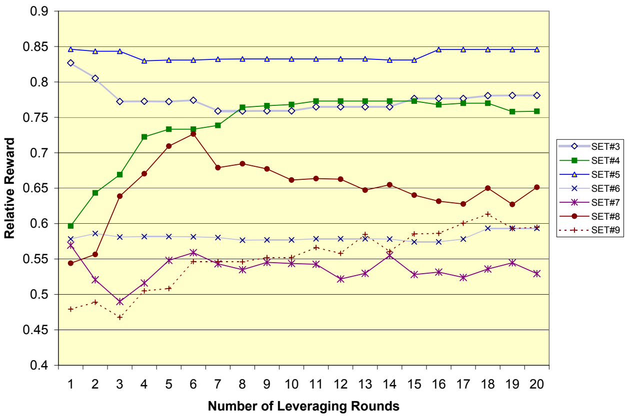
[1] The risk of overfitting increases with the complexity of the hypothesis.
[2] The control policy value was computed as the ratio of the cumulative reward collected by the resulting control policy to the optimal reward.
[3] Detailed and up-to-date results are available at [18].



(a) Sample errors on $Q(\cdot,\cdot)$

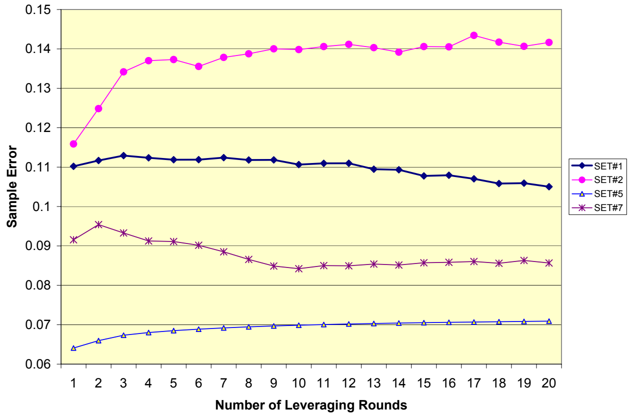


(b) Generalization errors on $Q(\cdot,\cdot)$

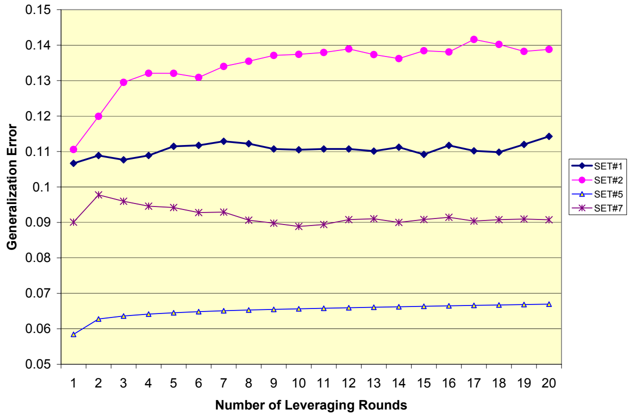


(c) Value of the resulting control policy

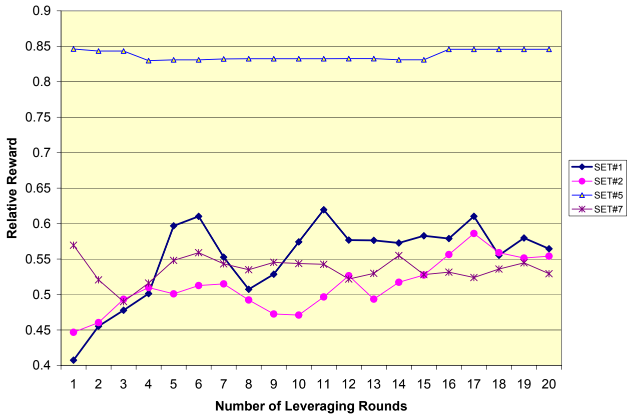
Figure 5. Experiments on MR ADORE using ResLev. Different combinations of features and neural network topologies were tried, and three performance measures (sample/generalization errors and average relative reward) were adopted.



(a) Sample errors on $Q(\cdot,\cdot)$



(b) Generalization errors on $Q(\cdot,\cdot)$



(c) Value of the resulting control policy

Figure 6. Contradictions among different measures exist in these experiments, which suggest the different behaviors between a regression problem and a decision-making problem in terms of boosting/leveraging methods applied to the value function.

fectiveness of RESLEV to that of other boosting/leveraging algorithms; (iv) developing new boosting/leveraging algorithms that directly optimize the *overall* decision-making performance as opposed to the regression error (such as sample/generalization errors) of the value function.

Contributions: In this paper we consider a state-of-the-art adaptive image interpretation system that casts vision as a non-discounted finite-horizon MDP control problem over a library of image processing operators. The key learning task in such systems is automated acquisition of the optimal control policy. We apply two recent leveraging algorithms, RESLEV and SQUARELEV.R, to learning the optimal value function. Experiments in the real-life domain of forest image interpretation indicate that (i) leveraging is indeed useful in improving complex sequential decision-making policies, and (ii) reducing the regression error in the value function can actually *decrease* the overall interpretation performance (i.e., the value of the resulting control policy). Consequently, the paper opens a novel research direction of developing boosting/leveraging algorithms that target reducing the control policy value directly.

Acknowledgements

Bruce Draper participated in the initial MR ADORE design stage. Terry Caelli, Li Cheng, Joan Fang, Ken Greenway, Dorothy Lau, Omid Madani, Rongzhou Man, David McNabb, and Guanwen Zhang have contributed in various ways. We are grateful for the funding from the Alberta Ingenuity Center for Machine Learning, the University of Alberta, and NSERC.

References

[1] M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, Cambridge, 1999.

[2] M. Arbib. *The metaphorical brain: An introduction to cybernetics as artificial intelligence and brain theory*. Wiley-Interscience, New York, 1972.

[3] M. Arbib. Segmentation, schemas, and cooperative computation. In S. Levin, editor, *MAA studies in Mathematics*, pages 118–155. 1978.

[4] A. G. Barto, S. J. Bradtke, and S. P. Singh. Learning to act using real-time dynamic programming. *Artificial Intelligence*, 72(1):81–138, 1995.

[5] V. Bulitko, L. Li, G. Lee, R. Greiner, and I. Levner. Adaptive image interpretation : A spectrum of machine learning problems. In *ICML-03 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, Washington D.C., August 2003.

[6] T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.

[7] B. Draper. From knowledge bases to Markov models to PCA. In *Workshop on Computer Vision System Control Architectures (held in conjunction with ICVS)*, Graz Austria, March 2003.

[8] B. Draper, J. Bins, and K. Baek. ADORE: Adaptive object recognition. In *International Conference on Vision Systems*, Spain, 2000.

[9] B. Draper, A. Hanson, and E. Riseman. Knowledge-directed vision: Control, learning and integration. In *Proceedings of the IEEE*, volume 84, pages 1625–1637, 1996.

[10] H. Drucker. Improving regressors using boosting techniques. In D. H. Fisher, editor, *Proc. Fourteenth International Conference on Machine Learning (ICML-97)*, pages 107–115. Morgan-Kaufmann, 1997.

[11] N. Duffy and D. Helmbold. Potential boosters? In K.-R. M. Sara Solla, Todd Leen, editor, *Advances in Neural Information Processing Systems (NIPS-99)*, volume 12, pages 258–264. MIT Press, 2000.

[12] N. Duffy and D. Helmbold. Boosting methods for regression. *Machine Learning*, 47:153–200, 2002.

[13] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[14] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML-96)*, pages 148–156, 1996.

[15] J. Friedman. Multivariate adaptive regression splines. *Annals of Statistics*, 19:1–141, 1991.

[16] J. Friedman. Greedy function approximation: A gradient boosting machine. Technical report, Department of Statistics, Stanford University, CA, March 2000.

[17] I. Levner, V. Bulitko, L. Li, G. Lee, and R. Greiner. Towards automated creation of image interpretation systems. In *Proceedings of the Sixteenth Australian Joint Conference on Artificial Intelligence (AI-03)*, Perth, Australia, December 2003.

[18] L. Li. Experimental results of RESLEV, 2003. Available at http://www.cs.ualberta.ca/∼lihong/research/reslev/reslev.htm.

[19] L. Li. ResLev: A leveraging method for regression. Technical report, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada, July 2003. Available at http://www.cs.ualberta.ca/∼lihong.

[20] R. Meir and G. Räsch. An introduction to boosting and leveraging. In S. Mendelson and A. Smola, editors, *Advanced Lectures on Machine Learning*, LNCS, pages 119–184. Springer, 2003.

[21] R. Rimey and C. Brown. Control of selective perception using Bayes nets and decision theory. *International Journal of Computer Vision*, 12:173–207, 1994.

[22] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[23] S. P. Singh and R. C. Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3):227–233, 1994.

[24] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, March 1998.

[25] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134 – 1142, Nov. 1984.