Active Learning with Statistical Models

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan
cohn@psyche.mit.edu, zoubin@psyche.mit.edu, jordan@psyche.mit.edu

Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

For many types of learners one can compute the statistically "optimal" way to select data. We review how these techniques have been used with feedforward neural networks [MacKay, 1992; Cohn, 1994]. We then show how the same principles may be used to select data for two alternative, statistically-based learning architectures: mixtures of Gaussians and locally weighted regression. While the techniques for neural networks are expensive and approximate, the techniques for mixtures of Gaussians and locally weighted regression are both efficient and accurate.

1 ACTIVE LEARNING – BACKGROUND

An *active* learning problem is one where the learner has the ability or need to influence or select its own training data. Many problems of great practical interest allow active learning, and many even require it.

We consider the problem of actively learning a mapping $X \rightarrow Y$ based on a set of training examples $\{(x_i, y_i)\}_{i=1}^{m}$, where $x_i \in X$ and $y_i \in Y$. The learner is allowed to iteratively select new inputs $\tilde{x}$ (possibly from a constrained set), observe the resulting output $\tilde{y}$, and incorporate the new examples $(\tilde{x}, \tilde{y})$ into its training set.

The primary question of active learning is how to choose which $\tilde{x}$ to try next. There are many heuristics for choosing $\tilde{x}$ based on intuition, including choosing places where we don't have data, where we perform poorly [Linden and Weber, 1993], where we have low confidence [Thrun and Möller, 1992], where we expect it to change our model [Cohn et al, 1990], and where we previously found data that resulted in learning [Schmidhuber and Storck, 1993].

In this paper we consider how one may select $\tilde{x}$ "optimally" from a statistical viewpoint. We first review how the statistical approach can be applied to neural networks, as described in MacKay [1992] and Cohn [1994]. We then consider two alternative, statistically-based learning architectures: mixtures of Gaussians and locally weighted regression. While optimal data selection for a neural network is computationally expensive and approximate, we find that optimal data selection for the two statistical models is efficient and accurate.

2 ACTIVE LEARNING – A STATISTICAL APPROACH

We denote the learner's output given input x as $\hat{y}(x)$. The mean squared error of this output can be expressed as the sum of the learner's bias and variance. The variance $\sigma_{\hat{y}}^2(x)$ indicates the learner's uncertainty in its estimate at x.[1] Our goal will be to select a new example $\tilde{x}$ such that when the resulting example $(\tilde{x}, \tilde{y})$ is added to the training set, the integrated variance IV is minimized:

$$IV = \int \sigma_{\hat{y}}^2 P(x) dx. \tag{1}$$

Here, $P(x)$ is the (known) distribution over X. In practice, we will compute a Monte Carlo approximation of this integral, evaluating $\sigma_{\hat{y}}^2$ at a number of random points drawn according to $P(x)$.

Selecting $\tilde{x}$ so as to minimize IV requires computing $\tilde{\sigma}_{\hat{y}}^2$, the new variance at x given $(\tilde{x}, \tilde{y})$. Until we actually commit to an $\tilde{x}$, we do not know what corresponding $\tilde{y}$ we will see, so the minimization cannot be performed deterministically.[2] Many learning architectures, however, provide an estimate of $P(\tilde{y}|\tilde{x})$ based on current data, so we can use this estimate to compute the *expectation* of $\tilde{\sigma}_{\hat{y}}^2$. Selecting $\tilde{x}$ to minimize the expected integrated variance provides a solid statistical basis for choosing new examples.

2.1 EXAMPLE: ACTIVE LEARNING WITH A NEURAL NETWORK

In this section we review the use of techniques from Optimal Experiment Design (OED) to minimize the estimated variance of a neural network [Fedorov, 1972; MacKay, 1992; Cohn, 1994]. We will assume we have been given a learner $\hat{y} = f_{\hat{w}}()$, a training set $\{(x_i, y_i)\}_{i=1}^m$ and a parameter vector $\hat{w}$ that maximizes a likelihood measure. One such measure is the minimum sum squared residual

$$S^2 = \frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}(x_i))^2 .$$

[1] Unless explicitly denoted, $\hat{y}$ and $\sigma_{\hat{y}}^2$ are functions of x. For simplicity, we present our results in the univariate setting. All results in the paper extend easily to the multivariate case.

[2] This contrasts with related work by Plutowski and White [1993], which is concerned with filtering an existing data set.

The estimated output variance of the network is

$$\sigma_{\hat{y}}^2 \approx S^2 \left(\frac{\partial \hat{y}(x)}{\partial w}\right)^T \left(\frac{\partial^2 S^2}{\partial w^2}\right)^{-1} \left(\frac{\partial \hat{y}(x)}{\partial w}\right)$$

The standard OED approach assumes normality and local linearity. These assumptions allow replacing the distribution $P(\tilde{y}|\tilde{x})$ by its estimated mean $\hat{y}(\tilde{x})$ and variance S^2. The expected value of the new variance, $\tilde{\sigma}_{\hat{y}}^2$, is then:

$$\langle \tilde{\sigma}_{\hat{y}}^2 \rangle \approx \sigma_{\hat{y}}^2 - \frac{\sigma_{\hat{y}}^2(x, \tilde{x})}{S^2 + \sigma_{\hat{y}}^2(\tilde{x})}, \text{ [MacKay, 1992].} \qquad (2)$$

where we define

$$\sigma_{\hat{y}}(x, \tilde{x}) \equiv S^2 \left(\frac{\partial \hat{y}(x)}{\partial w}\right)^T \left(\frac{\partial^2 S^2}{\partial w^2}\right)^{-1} \left(\frac{\partial \hat{y}(\tilde{x})}{\partial w}\right).$$

For empirical results on the predictive power of Equation 2, see Cohn [1994].

The advantages of minimizing this criterion are that it is grounded in statistics, and is optimal given the assumptions. Furthermore, the criterion is continuous and differentiable. As such, it is applicable in continuous domains with continuous action spaces, and allows hillclimbing to find the "best" $\tilde{x}$.

For neural networks, however, this approach has many disadvantages. The criterion relies on simplifications and strong assumptions which hold only approximately. Computing the variance estimate requires inversion of a $|w| \times |w|$ matrix for each new example, and incorporating new examples into the network requires expensive retraining. Paass and Kindermann [1995] discuss an approach which addresses some of these problems.

3 MIXTURES OF GAUSSIANS

The mixture of Gaussians model is gaining popularity among machine learning practitioners [Nowlan, 1991; Specht, 1991; Ghahramani and Jordan, 1994]. It assumes that the data is produced by a mixture of N Gaussians g_i, for $i = 1, ..., N$. We can use the EM algorithm [Dempster et al, 1977] to find the best fit to the data, after which the conditional expectations of the mixture can be used for function approximation.

For each Gaussian g_i we will denote the estimated input/output means as $\mu_{x,i}$ and $\mu_{y,i}$ and estimated covariances as $\sigma_{x,i}^2$, $\sigma_{y,i}^2$ and $\sigma_{xy,i}$. The conditional variance of y given x may then be written

$$\sigma_{y|x,i}^2 = \sigma_{y,i}^2 - \frac{\sigma_{xy,i}^2}{\sigma_{x,i}^2}.$$

We will denote as n_i the (possibly fractional) number of training examples for which g_i takes responsibility:

$$n_i = \sum_{j=1}^{m} \frac{P(x_j, y_j|i)}{\sum_{k=1}^{N} P(x_j, y_j|k)}.$$

For an input x, each g_i has conditional expectation $\hat{y}_i$ and variance $\sigma^2_{\hat{y},i}$:

$$\hat{y}_i = \mu_{y,i} + \frac{\sigma_{xy,i}}{\sigma^2_{x,i}}(x - \mu_{x,i}), \qquad \sigma^2_{\hat{y},i} = \frac{\sigma^2_{y|x,i}}{n_i}\left(1 + \frac{(x-\mu_{x,i})^2}{\sigma^2_{x,i}}\right).$$

These expectations and variances are mixed according to the prior probability that g_i has of being responsible for x:

$$h_i \equiv h_i(x) = \frac{P(x|i)}{\sum_{j=1}^N P(x|j)}.$$

For input x then, the conditional expectation $\hat{y}$ of the resulting mixture and its variance may be written:

$$\hat{y} = \sum_{i=1}^N h_i\,\hat{y}_i, \qquad \sigma^2_{\hat{y}} = \sum_{i=1}^N \frac{h_i^2 \sigma^2_{y|x,i}}{n_i}\left(1 + \frac{(x-\mu_{x,i})^2}{\sigma^2_{x,i}}\right).$$

In contrast to the variance estimate computed for a neural network, here $\sigma^2_{\hat{y}}$ can be computed efficiently with no approximations.

3.1 ACTIVE LEARNING WITH A MIXTURE OF GAUSSIANS

We want to select $\tilde{x}$ to minimize $\left\langle \tilde{\sigma}^2_{\hat{y}} \right\rangle$. With a mixture of Gaussians, the model's estimated distribution of $\tilde{y}$ given $\tilde{x}$ is explicit:

$$P(\tilde{y}|\tilde{x}) = \sum_{i=1}^N \tilde{h}_i P(\tilde{y}|\tilde{x}, i) = \sum_{i=1}^N \tilde{h}_i N(\hat{y}_i(\tilde{x}), \sigma^2_{y|x,i}(\tilde{x})),$$

where $\tilde{h}_i \equiv h_i(\tilde{x})$. Given this, calculation of $\left\langle \tilde{\sigma}^2_{\hat{y}} \right\rangle$ is straightforward: we model the change in each g_i separately, calculating its expected variance given a new point sampled from $P(\tilde{y}|\tilde{x}, i)$ and weight this change by $\tilde{h}_i$. The new expectations combine to form the learner's new expected variance

$$\left\langle \tilde{\sigma}^2_{\hat{y}} \right\rangle = \sum_{i=1}^N \frac{\tilde{h}_i^2 \left\langle \tilde{\sigma}^2_{y|x,i} \right\rangle}{n_i + \tilde{h}_i}\left(1 + \frac{(x - \tilde{\mu}_{x,i})^2}{\tilde{\sigma}^2_{x,i}}\right) \tag{3}$$

where the expectation can be computed exactly in closed form:

$$\left\langle \tilde{\sigma}^2_{y|x,i} \right\rangle = \left\langle \tilde{\sigma}^2_{y,i} \right\rangle - \frac{\left\langle \tilde{\sigma}^2_{xy,i} \right\rangle}{\tilde{\sigma}^2_{x,i}}, \qquad \left\langle \tilde{\sigma}^2_{y,i} \right\rangle = \frac{n\sigma^2_{y,i} + \tilde{h}_i \sigma^2_{y,i}(\tilde{x})}{n + \tilde{h}_i} + \frac{n\tilde{h}_i(\hat{y}_i(\tilde{x}) - \mu_{y,i})^2}{(n + \tilde{h}_i)^2},$$

$$\tilde{\mu}_{x,i} = \frac{n_i \mu_{x,i} + \tilde{h}_i \tilde{x}}{n_i + \tilde{h}_i}, \qquad \left\langle \tilde{\sigma}_{xy,i} \right\rangle = \frac{n\sigma_{xy,i}}{n + \tilde{h}_i} + \frac{n\tilde{h}_i(\tilde{x} - \mu_{x,i})(\hat{y}_i(\tilde{x}) - \mu_{y,i})}{(n + \tilde{h}_i)^2},$$

$$\tilde{\sigma}^2_{x,i} = \frac{n\sigma^2_{x,i}}{n + \tilde{h}_i} + \frac{n\tilde{h}_i(\tilde{x} - \mu_{x,i})^2}{(n + \tilde{h}_i)^2}, \qquad \left\langle \tilde{\sigma}^2_{xy,i} \right\rangle = \left\langle \tilde{\sigma}_{xy,i} \right\rangle^2 + \frac{n^2 \tilde{h}_i^2 \sigma^2_{y,i}(\tilde{x})(\tilde{x} - \mu_{x,i})^2}{(n + \tilde{h}_i)^4}.$$

4 LOCALLY WEIGHTED REGRESSION

We consider here two forms of locally weighted regression (LWR): kernel regression and the LOESS model [Cleveland et al, 1988]. Kernel regression computes $\hat{y}$ as an average of the y_i in the data set, weighted by a kernel centered at x. The LOESS model performs a linear regression on points in the data set, weighted by a kernel centered at x. The kernel shape is a design parameter: the original LOESS model uses a "tricubic" kernel; in our experiments we use the more common Gaussian

$$h_i(x) \equiv h(x - x_i) = \exp(-k(x - x_i)^2),$$

where k is a smoothing constant. For brevity, we will drop the argument x for $h_i(x)$, and define $n = \sum_i h_i$. We can then write the estimated means and covariances as:

$$\mu_x = \frac{\sum_i h_i x_i}{n},\ \sigma_x^2 = \frac{\sum_i h_i (x_i - x)^2}{n},\ \sigma_{xy} = \frac{\sum_i h_i (x_i - x)(y_i - \mu_y)}{n}$$
$$\mu_y = \frac{\sum_i h_i y_i}{n},\ \sigma_y^2 = \frac{\sum_i h_i (y_i - \mu_y)^2}{n},\ \sigma_{y|x}^2 = \sigma_y^2 - \frac{\sigma_{xy}^2}{\sigma_x^2}.$$

We use them to express the conditional expectations and their estimated variances:

$$\text{kernel:} \qquad \hat{y} = \mu_y, \qquad \sigma_{\hat{y}}^2 = \frac{\sigma_y^2}{n} \tag{4}$$

$$\text{LOESS:} \quad \hat{y} = \mu_y + \tfrac{\sigma_{xy}}{\sigma_x^2}(x - \mu_x), \quad \sigma_{\hat{y}}^2 = \frac{\sigma_{y|x}^2}{n}\left(1 + \frac{(x - \mu_x)^2}{\sigma_x^2}\right) \tag{5}$$

4.1 ACTIVE LEARNING WITH LOCALLY WEIGHTED REGRESSION

Again we want to select $\tilde{x}$ to minimize $\left\langle \tilde{\sigma}_{\hat{y}}^2 \right\rangle$. With LWR, the model's estimated distribution of $\tilde{y}$ given $\tilde{x}$ is explicit:

$$P(\tilde{y}|\tilde{x}) = N(\hat{y}(\tilde{x}), \sigma_{y|x}^2(\tilde{x}))$$

The estimate of $\left\langle \tilde{\sigma}_{\hat{y}}^2 \right\rangle$ is also explicit. Defining $\tilde{h}$ as the weight assigned to $\tilde{x}$ by the kernel, the learner's expected new variance is

$$\text{kernel: } \langle \tilde{\sigma}_{\hat{y}}^2 \rangle = \frac{\langle \tilde{\sigma}_y^2 \rangle}{n + \tilde{h}} \qquad \text{LOESS: } \langle \tilde{\sigma}_{\hat{y}}^2 \rangle = \frac{\left\langle \tilde{\sigma}_{y|x}^2 \right\rangle}{n + \tilde{h}}\left(1 + \frac{(x - \tilde{\mu}_x)^2}{\tilde{\sigma}_x^2}\right) \tag{6}$$

where the expectation can be computed exactly in closed form:

$$\left\langle \tilde{\sigma}_{y|x}^2 \right\rangle = \langle \tilde{\sigma}_y^2 \rangle - \frac{\langle \tilde{\sigma}_{xy}^2 \rangle}{\tilde{\sigma}_x^2}, \qquad \langle \tilde{\sigma}_y^2 \rangle = \frac{n\sigma_y^2 + \tilde{h}\sigma_y^2(\tilde{x})}{n + \tilde{h}} + \frac{n\tilde{h}(\hat{y}(\tilde{x}) - \mu_y)^2}{(n + \tilde{h})^2},$$
$$\tilde{\mu}_x = \frac{n\mu_x + \tilde{h}\tilde{x}}{n + \tilde{h}}, \qquad \langle \tilde{\sigma}_{xy} \rangle = \frac{n\sigma_{xy}}{n + \tilde{h}} + \frac{n\tilde{h}(\tilde{x} - \mu_x)(\hat{y}(\tilde{x}) - \mu_y)}{(n + \tilde{h})^2},$$
$$\tilde{\sigma}_x^2 = \frac{n\sigma_x^2}{n + \tilde{h}} + \frac{n\tilde{h}(\tilde{x} - \mu_x)^2}{(n + \tilde{h})^2}, \qquad \langle \tilde{\sigma}_{xy}^2 \rangle = \langle \tilde{\sigma}_{xy} \rangle^2 + \frac{n^2\tilde{h}^2\sigma_y^2(\tilde{x})(\tilde{x} - \mu_x)^2}{(n + \tilde{h})^4}.$$

5 EXPERIMENTAL RESULTS

Below we describe two sets of experiments demonstrating the predictive power of the query selection criteria in this paper. In the first set, learners were trained on data from a noisy sine wave. The criteria described in this paper were applied to predict how a new training example selected at point $\tilde{x}$ would decrease the learner's variance. These predictions, along with the actual changes in variance when the training points were queried and added, are plotted in Figure 1.

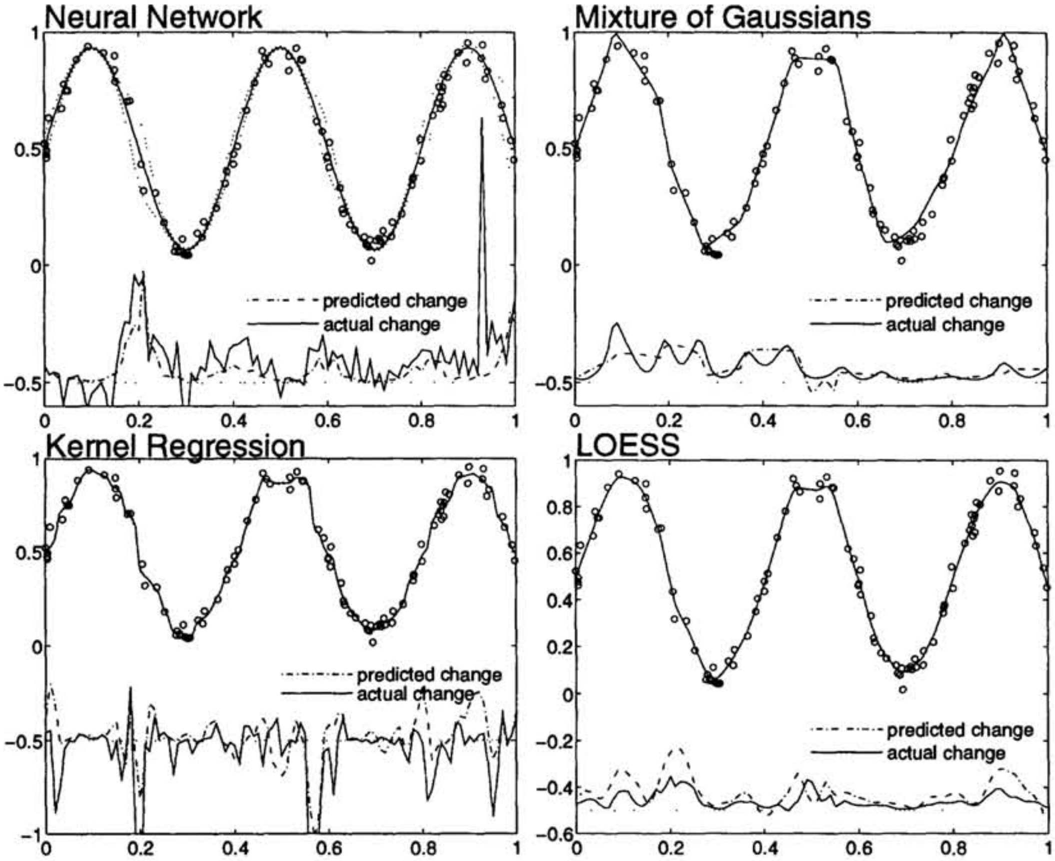

Figure 1: The upper portion of each plot indicates each learner's fit to noisy sinusoidal data. The lower portion of each plot indicates predicted and actual changes in the learner's average estimated variance when $\tilde{x}$ is queried and added to the training set, for $\tilde{x} \in [0, 1]$. Changes are not plotted to scale with learners' fits.

In the second set of experiments, we applied the techniques of this paper to learning the kinematics of a two-joint planar arm (Figure 2; see Cohn [1994] for details). Below, we illustrate the problem using the LOESS algorithm.

An example of the correlation between predicted and actual changes in variance on this problem is plotted in Figure 2. Figure 3 demonstrates that this correlation may be exploited to guide sequential query selection. We compared a LOESS learner which selected each new query so as to minimize expected variance with LOESS learners which selected queries according to various heuristics. The variance-minimizing learner significantly outperforms the heuristics in terms of both variance and MSE.

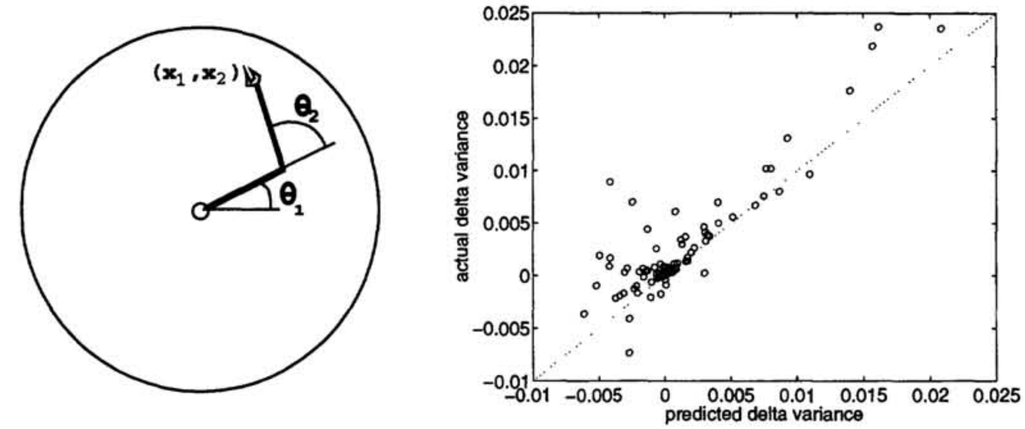

Figure 2: (left) The arm kinematics problem. (right) Predicted vs. actual changes in model variance for LOESS on the arm kinematics problem. 100 candidate points are shown for a model trained with 50 initial random examples. Note that most of the potential queries produce very little improvement, and that the algorithm successfully identifies those few that will help most.

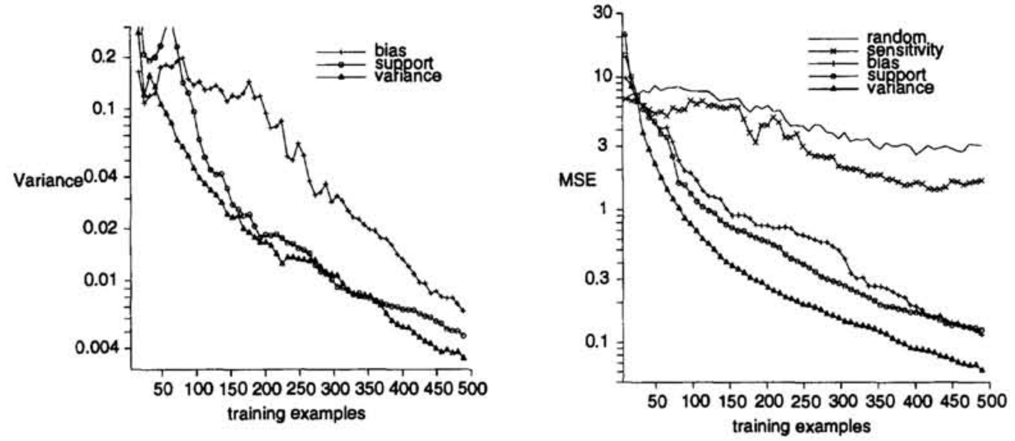

Figure 3: Variance and MSE for a LOESS learner selecting queries according to the variance-minimizing criterion discussed in this paper and according to several heuristics. "Sensitivity" queries where output is most sensitive to new data, "Bias" queries according to a bias-minimizing criterion, "Support" queries where the model has the least data support. The variance of "Random" and "Sensitivity" are off the scale. Curves are medians over 15 runs with non-Gaussian noise.

6 SUMMARY

Mixtures of Gaussians and locally weighted regression are two statistical models that offer elegant representations and efficient learning algorithms. In this paper we have shown that they also offer the opportunity to perform active learning in an efficient and statistically correct manner. The criteria derived here can be computed cheaply and, for problems tested, demonstrate good predictive power.

Acknowledgements

This work was funded by NSF grant CDA-9309300, the McDonnell-Pew Foundation, ATR Human Information Processing Laboratories and Siemens Corporate Research. We thank Stefan Schaal for helpful discussions about locally weighted regression.

References

W. Cleveland, S. Devlin, and E. Grosse. (1988) Regression by local fitting. *Journal of Econometrics* **37**:87–114.

D. Cohn, L. Atlas and R. Ladner. (1990) Training Connectionist Networks with Queries and Selective Sampling. In D. Touretzky, ed., *Advances in Neural Information Processing Systems 2*, Morgan Kaufmann.

D. Cohn. (1994) Neural network exploration using optimal experiment design. In J. Cowan et al., eds., *Advances in Neural Information Processing Systems 6*. Morgan Kaufmann.

A. Dempster, N. Laird and D. Rubin. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society Series B*, **39**:1–38.

V. Fedorov. (1972) *Theory of Optimal Experiments.* Academic Press, New York.

Z. Ghahramani and M. Jordan. (1994) Supervised learning from incomplete data via an EM approach. In J. Cowan et al., eds., *Advances in Neural Information Processing Systems 6*. Morgan Kaufmann.

A. Linden and F. Weber. (1993) Implementing inner drive by competence reflection. In H. Roitblat et al., eds., *Proc. 2nd Int. Conf. on Simulation of Adaptive Behavior*, MIT Press, Cambridge.

D. MacKay. (1992) Information-based objective functions for active data selection, *Neural Computation* **4**(4): 590–604.

S. Nowlan. (1991) Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures. CMU-CS-91-126, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Paass, G., and Kindermann, J. (1995). Bayesian Query Construction for Neural Network Models. *In this volume.*

M. Plutowski and H. White (1993). Selecting concise training sets from clean data. *IEEE Transactions on Neural Networks*, **4**, 305–318.

S. Schaal and C. Atkeson. (1994) Robot Juggling: An Implementation of Memory-based Learning. *Control Systems Magazine*, **14**(1):57–71.

J. Schmidhuber and J. Storck. (1993) Reinforcement driven information acquisition in nondeterministic environments. Tech. Report, Fakultät für Informatik, Technische Universität München.

D. Specht. (1991) A general regression neural network. *IEEE Trans. Neural Networks*, 2(6):568–576.

S. Thrun and K. Möller. (1992) Active exploration in dynamic environments. In J. Moody et al., editors, *Advances in Neural Information Processing Systems 4*. Morgan Kaufmann.