Learning Predictive Evaluation Function for the EM Algorithm *

CS281 Final Project, Prof. Stuart Russell

Shawn Chang
<shawnc@cs.berkeley.edu>

22 May 1998

Abstract

This paper shows by simple example that the solution space of the popular EM algorithm may contain many local optima of different qualities. Based on the previous works in local search algorithms, an algorithm is developed to learn a predictive evaluation function of state features, which not only provides a measure of goodness of a state, but also gives hints on how promising the state is if used as a starting state for a new EM run. Preliminary experiments show that the algorithm enhances the EM performance in some problem domains.

1 Introduction

In the past twenty years, the Expectation-Maximization (EM) algorithm has been successfully applied to a variety of problems involving incomplete data, such as learning hidden Markov models, training neural networks, and learning mixture models. The EM associates a given incomplete-data problem with a simpler complete-data problem, and iteratively finds the maximum likelihood estimates of the data. In a typical situation, the EM converges [McLachlan and Krishnan 97] monotonically to a fixed point in the state space, usually a local maximum.

Like the state space of many local search algorithms in global optimization problems, the state space for the EM in maximum likelihood estimation problems can be potentially very complex. In some domains, there exist many local maxima with very different likelihood estimates. Different starting states often lead to different local maxima. Figure 1 and 2 illustrate this problem with a simple example of learning a mixture of two univariate Gaussians to model a normalized Sine distribution with the EM.

Finding a global maxima in a complex state space has known to be difficult, especially in cases where large number of sub-optimal local maxima exist. Previous works [Boese et al. 94] have showed that in some combinatorial global optimization domains, there exist some apparent global geometric structures in the optimization cost surfaces. For example, when using hill-climbing in the Traveling Sales Person and Graph Bisection problems, the cost surfaces exhibit a global convex structure, "big valley", which suggests good

*Source codes and an HTML version of this report are available at http://www.cs.berkeley.edu/~ shawnc/281/finalprj

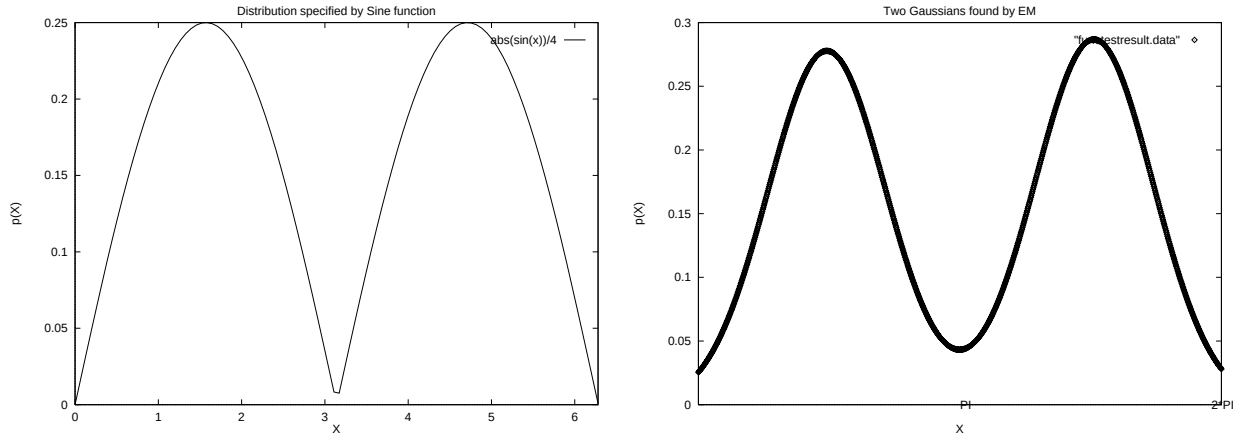

Figure 1: A Simple Example of Learning Mixture Model with EM: 1000 single-dimensional data points were generated randomly according to the distribution: $p(x) = abs(sin(x))/4$ (left). A mixture of two Gaussian distributions (right) was used to model this data set, and learned with the EM.

solutions are located near other good solutions in some predictable way.

In recent works [Boyan and Moore 98][Boyan 98], Boyan and Moore developed a strategy for learning to predict good starting states for local search algorithms in global optimization problems. In a global optimization problem, there exist a state space X and an objective function $Obj : X -> R$, and the goal is to to find a state $x*$ in X which maximizes Obj. Their STAGE algorithm defines an evaluation function of state features, which in addition to giving a measure of the utility of a state (directly related to the Obj), also predicts which states might lead to a good final state using a given local search algorithm. The evaluation function is updated periodically using the training data obtained from past local search trajectories. The STAGE then uses the evaluation function to suggest new promising starting states for the local search algorithm. The STAGE has been demonstrated to be very effective when used with local search algorithms like hill-climbing, WALKSAT, in several large real-world applications. Can we apply this idea to the EM?

There are many similarities between the EM and the local search algorithms used in the global optimization problems. Applying the EM in an incomplete-data problem can actually be perceived as a special instance of local search. For example, when learning a mixture of Gaussians, the state space consists of all possible assignments to the means, covariance matrices, and prior probabilities of the Gaussian distributions. The EM algorithm starts with some assignment to these state variables, and iteratively updates the state variables until it converges to a state with a locally maximum likelihood estimate of training data. The EM iterations correspond directly to the trajectories produced by a local search algorithm moving around in the neighborhood structure of a global optimization problem.

[Dempster et al. 77] shows that convergence of the EM is linear with the rate of convergence proportional to λ_{max}, where λ_{max} is the maximal fraction of missing information. This implies that the EM can be slow

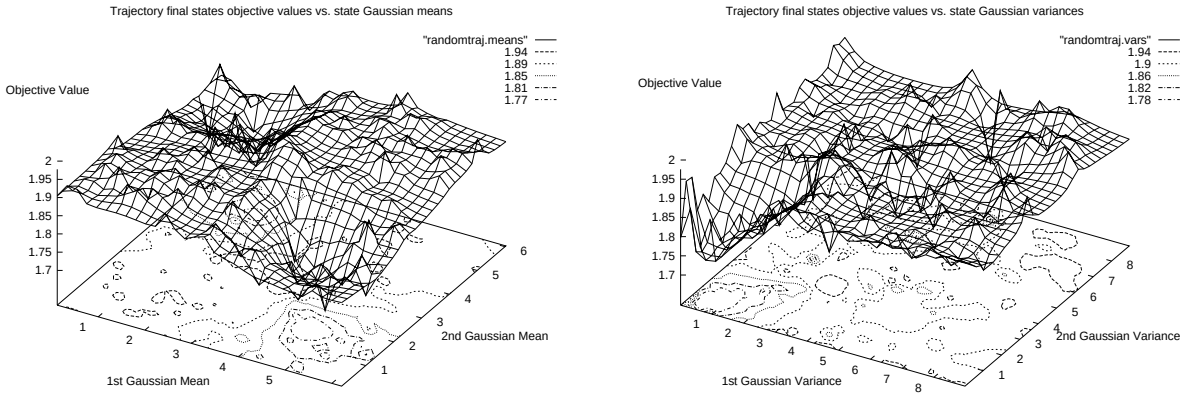

Figure 2: Surface Plots of Minimum Negative Log Likelihood Estimates Found by EM vs. Starting States: For the example in Figure 1, three hundred EM runs were started from randomly generated states, and the minimum negative log likelihood estimates of the training data were recorded. Each run was allowed to continue until the EM made no progress in the past ten iterations.

to converge, and too many re-starts are probably not desired. When the solution space has many sub-optimal fixed points, knowing which states are promising as starting state will certainly help.

As illustrated in Figure 2, the local extrema are abundant in the solution space of the EM, and the qualities of these extrema may differ significantly. Only some of these extrema give near-global-optimal likelihood estimates. Although it might not be easily identifiable, there may possibly exist some relevance between certain features (computed from the state descriptions) of a starting state to the solution found by the EM. Hence, it is reasonable to hope a STAGE-like algorithm will enhance the performance of the EM. The monotonic and deterministic characteristics of ordinary EM trajectories also help applying the STAGE to the EM. Each state along an EM trajectory can be directly used as a training data point for the predictive evaluation function; whereas in some stochastic local search algorithms, most states along a trajectory except the final state needs to be discounted, and reinforcement learning techniques are often necessary.

The EM and its extensions actually consist of a large family of algorithms, usually specific to each application. To demonstrate that the EM can benefit from the learning of a predictive evaluation function, this project focuses on applying the EM algorithm in learning a mixture of Gaussian distributions. The Gaussian distribution has many nice properties [Bishop 96]. In particular, the EM formulation of the mixture of Gaussians provides closed form update formula for the state variables, the means, covariance matrices, and the prior probabilities. The [Bishop 96] gives update formulas for a special case of multi-dimensional Gaussians, where each has a covariance matrix which is some scalar multiple of the identity matrix. In the general case, the update formulas for a mixture of multi-variate Gaussians with arbitrary covariance matrices are as follows:

$$\mu_j^{new} = \frac{\sum_{i=1}^{N} z_{ji}^{old} \mathbf{x}_i}{\sum_{i=1}^{N} z_{ji}^{old}} \tag{1}$$

$$\mathbf{\Sigma}^{new} = \frac{\sum_{i=1}^{N} z_{ji}^{old} (\mathbf{x}_i - \mu_j^{new})(\mathbf{x}_i - \mu_j^{new})_T}{\sum_{i=1}^{N} z_{ji}^{old}} \tag{2}$$

and

$$P_j^{new} = \frac{1}{N} \sum_{i=1}^{N} z_{ji}^{old} \tag{3}$$

where

$$z_{ji} = Prob(j|\mathbf{x}^i) \tag{4}$$

It is possible to also consider the number of Gaussians in a mixture model as another variable for optimization. However, for simplicity, this work assumes this number is predetermined, possibly by cross-validation.

This project attempts to develop a STAGE-like algorithm to enhance the performance of the EM by learning to predict which starting states lead to good local optima. The next section describes the algorithm in full detail. Section 3 presents some preliminary experimental results. Finally, we will conclude with discussions and future works.

2 Algorithm

The algorithm presented here follows closely to the framework of the STAGE. For convenience, we will call it STAGE-EM hereafter. We first give an overview of the STAGE-EM; we then discuss in detail learning and using the predictive evaluation function. We also give a simple illustrative example of an actual STAGE-EM run.

2.1 Overview

In learning a mixture of Gaussians using the EM, we are to find an optimal assignment to the means, covariance matrices, and prior probabilities of the Gaussian components, which gives the maximum likelihood estimate of the training data. For convenience, we use the negative log of the likelihood estimate as the objective value (Obj). Hence, the task is to find a state with minimum Obj. In addition, we define another quantity $V(x)$ as the expected best Obj on a trajectory that starts from a state x using the EM. We also refer to $V(x)$ as the predictive evaluation function, whose format will be described shortly.

Figure 3 gives a simple flowchart of the STAGE-EM. In the beginning of the first iteration, a random starting state is generated. Then, the STAGE-EM repeats the following sequence: running EM to optimize Obj, training the predictive evaluation function $V(x)$ with the new EM trajectory, local searching in the state space to optimize $V(x)$ and producing a new starting state for the EM. The loops end when a predefined number of iterations is reached. In each iteration, the EM uses a newly found starting state suggested by the $V(x)$, unless the local search on $V(x)$ makes no progress. In that case, a new starting state is randomly generated.

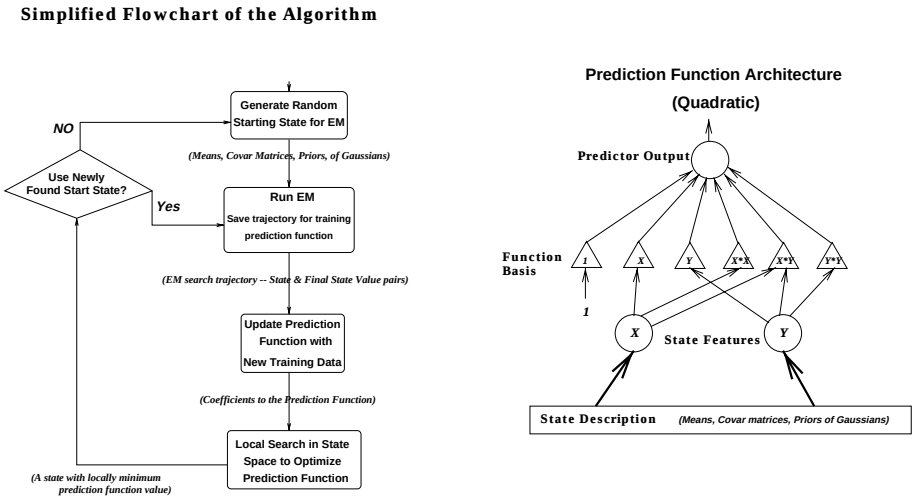

Figure 3: STAGE-EM Algorithm: flowchart (left) and predictive evaluation function architecture (right).

2.2 The Predictive Evaluation Function

The predictive evaluation function $V(x)$ plays a central role in the STAGE-EM. The $V(x)$ is a function of some state features. A state feature can be any deterministic function computed from a given state description. Often used features include the variances of some state variables, the means of some state variables, and some components of the Obj. The format of the function is problem dependent. It can be a simple linear regression, a quadratic regression, or even a multi-layer perceptron. However, overly complex functions have problems of over-fitting and may require large amount of training data. Conversely, very simple functions may not capture the state space structure. The state features are not limited to the state variables.

In this work, we use a simple quadratic predictive evaluation function. This evaluation function resembles the appearance of a radial basis function network, as shown in Figure 3. Some state features are extracted from a state description; a number of basis functions of up to second degree are formed from the features; a linear combination of the basis functions gives the final prediction. The training task is thus to learn the coefficients of the basis functions. This is done using an on-line linear regression, which gives the same result as off-line batch training, but does not require storing all past training data points. Because of the problems with singular matrices, the singular value decomposition (SVD) technique is used. Note that in STAGE, for certain stochastic search algorithms, the linear regression can be easily modified to take into account of a discount factor, and hence resembles a $TD(\lambda)$ reinforcement learning method.

Once a predictive evaluation function is trained, it can be used to predict how promising each state is as a starting state for the EM. To find a good starting state, local search is performed on the evaluation function to optimize $V(x)$. It should be noted that although the evaluation function is only in a quadratic form of the

state features, it is generally not possible to find the global optimum by simply taking some partial derivatives. The features are usually complex functions of the state variables and training data, and they are often not invertible. Therefore, finding a global optimum of the $V(x)$ in terms of state features does not actually give a new starting state. Any local search algorithm can be used here, and simple ones like stochastic hill-climbing are often good choice. An interesting note here is that, we can actually use the STAGE (or STAGE-EM) for this local search task, and the whole algorithm becomes recursive in some sense. However, for most problems, this second level application of STAGE might require a large amount of training data to be effective.[1]

2.3 A Simple Example

To illustrate the operations of the STAGE-EM, we again use the example given in Figure 1 and 2. In this task, one thousand single dimensional data points were randomly generated according to a distribution, $p(x) = abs(sin(x))/4$, in the interval, $0 \leq x \leq 2\pi$. A mixture of two univariate Gaussians were used to model this distribution, and hence there were six state variables, the means, the variances, and the prior probabilities of the two Gaussians. The selected features were the variance of the two means, which dictates the distance between the centers of the Gaussians, and the variance of the two variances, which dictates the difference in the widths of the two Gaussians. Let us refer to the two features, p and q, and coefficients $v0...v5$. Then, the predictive evaluation function can then be written as:

$$V(x) = v_0 * 1 + v_1 * p + v_2 * q + v_3 * p^2 + v_4 * p * q + v_5 * q^2 \tag{5}$$

Figure 5 (right) shows the best negative log likelihood estimates of twelve successive runs of the Random Multi-restart EM, and twelve successive runs of the STAGE-EM, using the same random number seed. Figure 4 and Figure 5 (left) shows the learned predictive evaluation function surface plots for the STAGE-EM at the end of EM runs 2, 6 and 9. It can be observed that the STAGE-EM learned overtime that the good starting states lie in two regions with low predicted *obj* values. Both the two regions agree on one of the state feature, the variance of the means, but differ in the other state feature. However, as most trajectories ended up at a state closer to one of the region, the local search naturally found a new starting state in that region. Eventually a fairly good state was found in EM run 10 for the STAGE-EM.

3 Experiments

This section presents some preliminary experimental results of applying the STAGE-EM to mixture model learning problems. Due to time and resource constraints, four experiments were conducted only on artificially generated data sets. From these simple experiments, we hope to gain some insights on the operation of the STAGE-EM, and whether it is promising to extend the algorithm to other applications.

The training and testing data sets in each experiment were generated with the same data generating function. Each set contains 2000 data points. The top table in Table 1 shows the description of the experiments. For each experiment, the STAGE-EM was run against a random multi-restart EM with the same random number seed. The total computation time consumed by each algorithm in each trial was held approximately equal. Each trial was repeated 30 times under the same condition. The tabulated results in the bottom table in Table 1 show the percentage improvements of the STAGE-EM over random multi-restart EM on the

[1] This work has included facilities for this recursive call, but no experiment has been conducted on it.

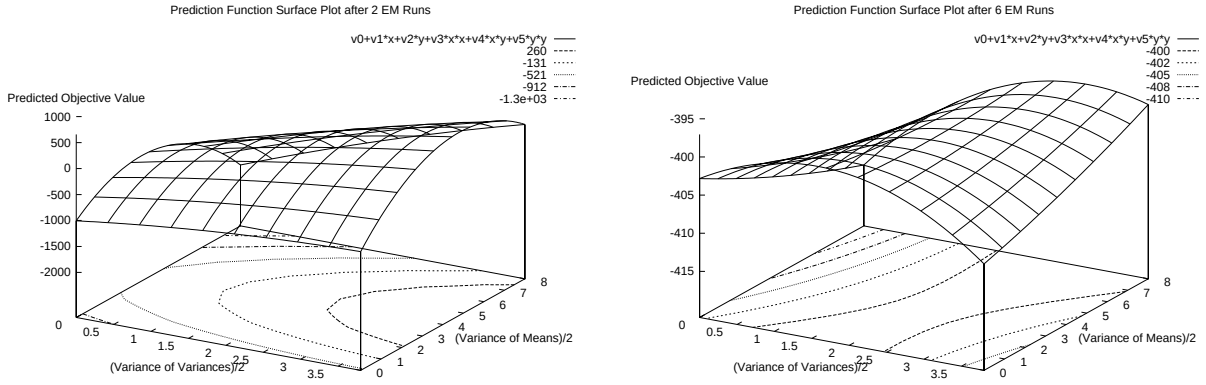

Figure 4: Sample Run of the STAGE-EM for the Example in Figure 1: Predictive Evaluation Function surface plots after 2 (left) and 6 (right) EM runs.

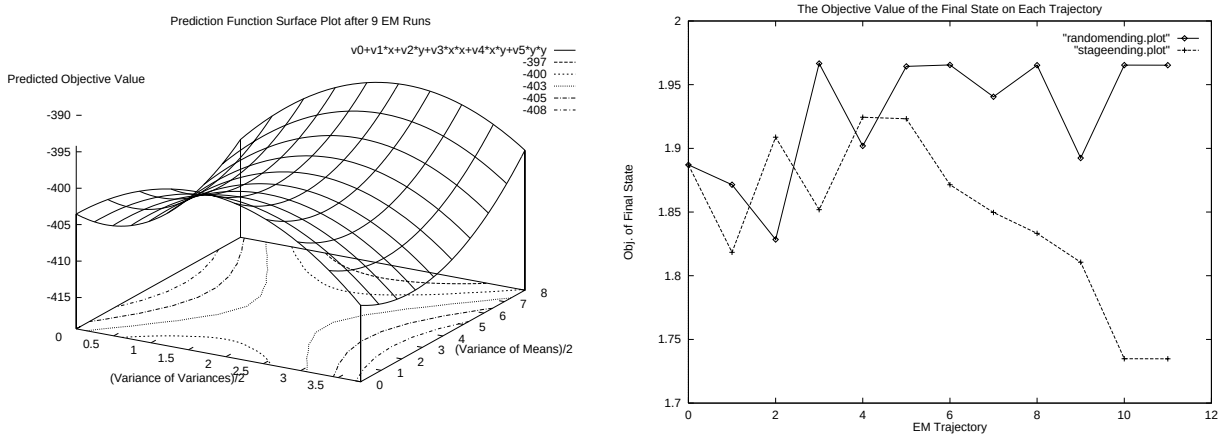

Figure 5: Sample Run of the STAGE-EM for the Example in Figure 1: On the left, a Predictive Evaluation Function surface plot after 9 EM restarts. On the right, plots of minimum negative log likelihood estimates found by Random Multi-restart EM (solid line) and STAGE-EM (broken line) at consecutive EM runs.

Task	Dimension	Data Generating Function	Number of Gaussians
1	3	$z = sin(x)tan(y)$	5
2	3	$z = (x^3 + y^3 + 5xy^2 + 7x^2y + 9x^2 + 11y^2 + 13xy + 15x + 17y + 1)/1000$	7
3	3	$z = sin(xy/10)(x + y)$	10
4	4	$z = sin(x)tan(y)cos(z)$	5

Task	Percentage Improvement on Maximum Likelihood Estimates					
	Training Set			Test Set		
	Average	Best	Worst	Average	Best	Worst
1	5.51	10.66	-1.11	5.38	9.28	-2.17
2	2.04	3.72	-1.50	2.04	5.16	-2.25
3	5.27	10.90	0.05	5.74	11.61	2.43
4	2.25	4.07	0.07	1.60	4.36	-1.78

Table 1: Experiment Results: The top table shows the description of each task. The bottom table shows the percentage improvements of STAGE-EM over Random Multi-restart EM for each task in 30 trials. The improvement calculation was based on the (geometric) average of the maximum likelihood estimates of all data points.

(geometric) average likelihood estimates of both training set and testing set data points. The overall results show modest improvements were achieved by the STAGE-EM. In some cases, the improvements were over ten percent, and in a few rare cases, the STAGE-EM performed actually worse.

For this set of experiments, two state features were used for constructing the predictive evaluation function, the variance of average widths of the Gaussians, and the negative log likelihood estimates (the Obj itself) on partial training data. In most cases, the learned evaluation function exhibits similar surface shapes, where the predicted good starting states lie in the region of high variance of widths of the Gaussians, and low Obj. Figure 6 (left) shows the predictive evaluation function surface plot obtained in one of the runs for Task 1. A learning curve for Task 1 has also been generated for various sizes of training set, also shown in Figure 6 (right).

4 Discussion and Future Works

The preliminary experimental results described in the previous section show that the STAGE-EM may enhance the performance of the EM algorithm in learning mixture models of Gaussian distributions. However, whether a problem can benefit from the STAGE-EM, and by how much if it can, is largely problem and user dependent. When facing a new problem, it is preferable to first run the EM algorithm a few times and examine the results analytically. Typically, the STAGE-EM is most promising in domains where the EM performs reasonably well, but still exhibits significant differences in the qualities of solutions when started from different state configurations. One important criterion is the existence of a coherent global structure of solution space associated with some state features. However, in many cases, such a global structure is not easily identifiable. The discovery of this global structure depends on the state features selected, the predictive evaluation function format adopted, and the regression techniques used.

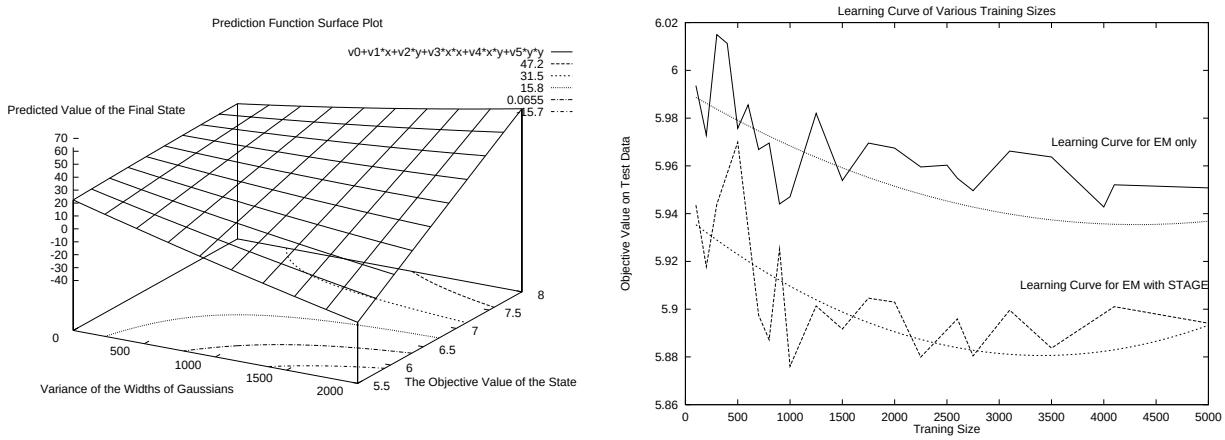

Figure 6: Plots for Task 1: On the left, the predictive evaluation function surface plot learned by STAGE-EM. On the right, learning curves (and their quadratic fits) of the Random Multi-restart EM (top) and the STAGE-EM for various training set sizes.

A few problems remain with the current formulation of the STAGE-EM algorithm (and the STAGE algorithm in general). Which state features are selected for constructing the predictive evaluation function plays a vital role in the performance of the STAGE-EM. Although there are a few suggestions for possibly good features, such as using the variance of certain state variables, the feature selection remains a black art. One may hope to recognize some general pattern of useful state features once the algorithm is applied to more problem domains. Another difficulty for applying the STAGE-EM is that, for each new problem, a large number of parameters need to be tuned in order to achieve any reasonable performance. The tunable parameters include the number of state features, the step size and patience for the local search used to optimize the evaluation function, and so forth, plus many parameters associated with the EM formulation itself. Correct settings of these parameters require much experience, domain knowledge, and computation resources.

Despite the existing problems with the current formulation of the STAGE-EM, the wide usage and applicability of the EM algorithm indicates that there may still exist many potential applications for the STAGE-EM. It would be of both theoretical and practical interests to adapt the the STAGE-EM to other domains where the EM algorithm succeeds, such as learning HMMs, learning Bayesian network structures, and a variety of other incomplete-data problems.

References

[Bishop 96] Bishop, C. M. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, England, 1996.

[Boese et al. 94] Boese, K. D.; Kahng, A. B.; and Muddu, S. 1994. A new adaptive multi-start technique for combinatorial global optimizations. *Operations Research Letters* 16:101-113.

[Boyan 98] Boyan, J. A. "Learning Evaluation Functions for Global Optimizations." *Ph.D. Thesis (draft)*, CMU, May 1998.

[Boyan and Moore 98] Boyan, J. A. and A. W. Moore. "Learning Evaluation Functions for Global Optimization and Boolean Satisfiability." *Fifteenth National Conference on Artificial Intelligence (AAAI), 1998* (to appear).

[Dempster et al. 77] Dempster, A.P.; Laird, N. M.; and Rubin, D. B. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society.* Series B, 39(1), 1-38.

[McLachlan and Krishnan 97] McLachlan, G. J. and T. Krishnan. *The EM Algorithm and Extensions.* John Wiley & Sons, Inc. New York, NY, 1997.