In M. C. Mozer, P. Smolensky, D. S. Touretzky, J. L. Elman, & A. S. Weigend (eds.), *Proceedings of the 1993 Connectionist Models Summer School.* pp. 316–323. Hillsdale, NJ: Erlbaum Associates, 1994. Email to: zoubin@psyche.mit.edu.

Solving inverse problems using an EM approach to density estimation

Zoubin Ghahramani[1]
Department of Brain & Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139
zoubin@psyche.mit.edu

This paper proposes density estimation as a feasible approach to the wide class of learning problems where traditional function approximation methods fail. These problems generally involve learning the inverse of causal systems, specifically when the inverse is a non-convex mapping. We demonstrate the approach through three case studies: the inverse kinematics of a three-joint planar arm, the acoustics of a four-tube articulatory model, and the localization of multiple objects from sensor data.

The learning algorithm presented differs from regression-based algorithms in that no distinction is made between input and output variables; the joint density is estimated via the EM algorithm and can be used to represent any input/output map by forming the conditional density of the output given the input.

Causality in physical systems induces directionality in the relations between variables measured from them. Thus, one can generally define a forward and an inverse direction of mapping. The forward direction is the causal direction, for example, from the forces applied to an object to the motion outcome, from the joint angles of an arm to the Cartesian coordinate of the finger, or from the configuration of a vocal tract to the sound frequencies produced. Similarly, the inverse direction is the non-causal direction. If the goal is to control the physical system the the inverse direction of mapping is particularly relevant. Returning to the above examples, this is the mapping from desired motion of an object to the forces required, from desired Cartesian finger coordinates to required joint angles, or from desired sound frequencies to required vocal tract configuration. In general the forward direction will be a function, whereas the inverse direction may be one-to-many and therefore not a function.

One-to-many relations are often difficult to learn with function approximation methods. This difficulty arises from the fact that if the image of an input is a non-convex region in the output, then the least-squares solution may fall outside this region (for further discussion of non-convexity see [12]).

This paper proposes density estimation as a feasible approach to the wide class of non-convex learning problems where function approximation and non-linear regression methods fail. The learning algorithm presented here differs from regression-based algorithms in that no distinction is made between input and output variables; the joint density is estimated and this estimate can then be used to form any input/output map. Thus, to estimate the vector function $\mathbf{y} = f(\mathbf{x})$ the joint density $P(\mathbf{x}, \mathbf{y})$ is estimated and, given a particular input $\mathbf{x}$, the conditional density $P(\mathbf{y}|\mathbf{x})$ is formed. If a single estimate of $\mathbf{y}$ is desired rather than the full conditional density, several methods can be applied. For example, the estimate can be set to $\hat{\mathbf{y}} = E(\mathbf{y}|\mathbf{x})$, the expectation of $\mathbf{y}$ given $\mathbf{x}$.

In particular, the density estimation algorithm presented is based on maximizing the likelihood of a parametric mixture model using the EM algorithm [3]. This approach provides a single framework for real, discrete, or mixed data, and generalizes naturally to data sets with arbitrary missing data patterns. In

[1]This research was supported by a grant from the McDonnell-Pew Foundation. This paper would not have been possible without the support and helpful comments of Michael I. Jordan and his research group. Thanks to Geoff Hinton for his insightful comments and review and to John F. Houde for the data from the four-tube model of the vocal tract.

principle any density estimation algorithm (e.g. [16, 17]) could be used within this framework for solving inverse problems. However, the parametric mixture models presented here benefit from the simple form of their conditional densities, the convergence speed of the EM algorithm, and a principled way for dealing with missing data.

Density Estimation using EM

General Theory

This section outlines the learning algorithm for mixture models [3, 4, 15]. We assume that the data $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ were generated independently by a mixture density:

$$P(\mathbf{x}_i) = \sum_{j=1}^{M} P(\mathbf{x}_i|\omega_j;\theta_j)P(\omega_j), \tag{1}$$

where each component of the mixture is denoted ω_j and parametrized by θ_j. Thus the log of the likelihood of the parameters given the data set is

$$l(\theta|\mathcal{X}) = \log \prod_{i=1}^{N} \sum_{j=1}^{M} P(\mathbf{x}_i|\omega_j;\theta_j)P(\omega_j) = \sum_{i=1}^{N} \log \sum_{j=1}^{M} P(\mathbf{x}_i|\omega_j;\theta_j)P(\omega_j). \tag{2}$$

We seek to find the parameter vector that maximizes $l(\theta|\mathcal{X})$. However, this function is not easily maximized numerically because it involves the log of a sum. Intuitively it is not easily maximized because for each data point there is a "credit-assignment" problem, i.e. it is not clear which component of the mixture generated that data point and thus which parameters to adjust to fit that data point.

The EM algorithm applied to mixtures is an iterative method for overcoming this credit-assignment problem. The intuition behind it is that if one had access to a "hidden" random variable $\mathbf{z}$ that indicated which data point was generated by which component, then the maximization problem would decouple into a set of simple maximizations. Mathematically, given $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ a "complete-data" log likelihood function could be written,

$$l_c(\theta|\mathcal{X}, \mathcal{Z}) = \sum_{i=1}^{N} \sum_{j=1}^{M} z_{ij} \log P(\mathbf{x}_i|\mathbf{z}_i;\theta)P(\mathbf{z}_i;\theta), \tag{3}$$

such that it does not involve a log of a summation.

As proven in [3], $l(\theta|\mathcal{X})$ can be maximized by iterating the following two steps,

$$\begin{array}{lrcl} \text{E step:} & Q(\theta|\theta_k) & = & E[l_c(\theta|\mathcal{X}, \mathcal{Z})|\mathcal{X}, \theta_k] \\ \text{M step:} & \theta_{k+1} & = & \arg\max_\theta Q(\theta|\theta_k). \end{array} \tag{4}$$

The E (Expectation) step computes the expected complete data log likelihood and the M (Maximization) step finds the parameters that maximize this likelihood.

Real case: mixture of Gaussians

Real valued data will be modeled as generated by a mixture of Gaussians. For this model the E-step simplifies to computing $h_{ij} \equiv E[z_{ij}|\mathbf{x}_i, \theta_k]$, the probability that Gaussian j, as defined by the mean $\hat{\boldsymbol{\mu}}_j$ and covariance matrix $\hat{\Sigma}_j$ estimated at time step k, generated data point i:

$$h_{ij} = \frac{|\hat{\Sigma}_j^k|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j^k)^T \hat{\Sigma}_j^{k,-1}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j^k)\}}{\sum_{l=1}^{M} |\hat{\Sigma}_l^k|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_l^k)^T \hat{\Sigma}_l^{k,-1}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_l^k)\}}. \tag{5}$$

The M-step then involves re-estimating the means and covariances of the Gaussians using the data set weighted by the h_{ij}:

$$\hat{\boldsymbol{\mu}}_j^{k+1} = \frac{\sum_{i=1}^{N} h_{ij}\mathbf{x}_i}{\sum_{i=1}^{N} h_{ij}}, \qquad \hat{\Sigma}_j^{k+1} = \frac{\sum_{i=1}^{N} h_{ij}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j^{k+1})(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j^{k+1})^T}{\sum_{i=1}^{N} h_{ij}}. \tag{6}$$

Discrete case: mixture of Bernoullis

D-dimensional binary data $\mathbf{x} = (x_1, \ldots, x_d \ldots x_D)$, $x_d \in \{0, 1\}$, will be modeled as generated by a mixture of m Bernoulli densities. That is,

$$P(\mathbf{x}|\theta) = \sum_{j=1}^{M} P(\omega_j) \prod_{d=1}^{D} \mu_{j\,d}^{x_d} (1 - \mu_{j\,d})^{(1-x_d)}. \tag{7}$$

For this model the E-step and M-step are:

$$\text{E step:} \quad h_{ij} = \frac{\prod_{d=1}^{D} \hat{\mu}_{j\,d}^{x_{i\,d}} (1 - \hat{\mu}_{j\,d})^{(1-x_{i\,d})}}{\sum_{l=1}^{M} \prod_{d=1}^{D} \hat{\mu}_{l\,d}^{x_{i\,d}} (1 - \hat{\mu}_{l\,d})^{(1-x_{i\,d})}}, \qquad \text{M step:} \quad \hat{\boldsymbol{\mu}}_j^{k+1} = \frac{\sum_{i=1}^{N} h_{ij} \mathbf{x}_i}{\sum_{i=1}^{N} h_{ij}}. \tag{8}$$

More generally, discrete or categorical data can be modeled as generated by a mixture of multinomial densities and similar derivations for the learning algorithm can be applied. Moreover, the extension to data with mixed real, binary, and categorical dimensions can also be readily derived.

The EM algorithm has traditionally been used in statistics for two distinct applications: to estimate the parameters of mixture models, as shown here, and to deal with arbitrary patterns of missing values in the data. A combination of both these applications of the EM algorithm, resulting in a general learning algorithm for incomplete data, is presented in [7].

Supervised learning

The above sections have outlined the learning algorithms for estimating a mixture density from a data set. When viewed as supervised learning each vector $\mathbf{x}_i$ in the training set is composed of an "input" subvector $\mathbf{x}_i^{\mathrm{i}}$ and a "target" or output subvector $\mathbf{x}_i^{\mathrm{o}}$. Applying the learning algorithm we obtain an estimate of the density of the data in this input/output space. For the Gaussian mixture case this estimate can be used to approximate a function in the following way:

Given the input vector $\mathbf{x}_i^{\mathrm{i}}$ we extract all the relevant information from the joint p.d.f. $P(\mathbf{x}^{\mathrm{i}}, \mathbf{x}^{\mathrm{o}})$ by conditionalizing to $P(\mathbf{x}^{\mathrm{o}}|\mathbf{x}_i^{\mathrm{i}})$. For a single Gaussian this conditional density is normal, and by linearity, since $P(\mathbf{x}^{\mathrm{i}}, \mathbf{x}^{\mathrm{o}})$ is a mixture of Gaussians so is $P(\mathbf{x}^{\mathrm{o}}|\mathbf{x}^{\mathrm{i}})$. In principle, this conditional density is the final output of the density estimator. That is, given a particular input the network returns the complete conditional density of the output. However, for the purposes of comparison to function approximation methods and since many applications require a single estimate of the output, we will outline three possible ways to obtain such an estimate $\hat{\mathbf{x}}$ of $\mathbf{x}^{\mathrm{o}} = f(\mathbf{x}_i^{\mathrm{i}})$:

- Least squares estimate (LSE) takes $\hat{\mathbf{x}}^{\mathrm{o}}(\mathbf{x}_i^{\mathrm{i}}) = E(\mathbf{x}^{\mathrm{o}}|\mathbf{x}_i^{\mathrm{i}})$;
- Stochastic Sampling (STOCH) samples according to the distribution $\hat{\mathbf{x}}^{\mathrm{o}}(\mathbf{x}_i^{\mathrm{i}}) \sim P(\mathbf{x}^{\mathrm{o}}|\mathbf{x}_i^{\mathrm{i}})$;
- Single component LSE (SLSE) takes $\hat{\mathbf{x}}^{\mathrm{o}}(\mathbf{x}_i^{\mathrm{i}}) = E(\mathbf{x}^{\mathrm{o}}|\mathbf{x}_i^{\mathrm{i}}, \omega_j)$ where $j = \arg\max_k P(z_k|\mathbf{x}_i^{\mathrm{i}})$. That is, for a given input, SLSE picks the Gaussian with highest posterior, and for that Gaussian approximates the output with the LSE estimator given by that Gaussian alone.

Looking more closely at the LSE estimator we note that we can write it as

$$\hat{\mathbf{x}}^{\mathrm{o}}(\mathbf{x}_i^{\mathrm{i}}) = \frac{\sum_{j=1}^{M} h_{ij} [\boldsymbol{\mu}_j^{\mathrm{o}} + \Sigma_j^{\mathrm{oi}} \Sigma_j^{\mathrm{oo}^{-1}} (\mathbf{x}_i^{\mathrm{i}} - \boldsymbol{\mu}_j^{\mathrm{i}})]}{\sum_{j=1}^{M} h_{ij}}, \tag{9}$$

from which we see that the LSE function estimate is a weighted sum of linear approximations, where the weights h_{ij} vary nonlinearly over the input space. In fact, the LSE estimator on a Gaussian mixture has interesting relations to algorithms such as CART [2], MARS [6], and competitive modular networks [11], as the mixture of Gaussians competitively partitions the input space, and learns a linear regression surface on each partition (details are given in [7]). In the limit, as the covariance matrices go to zero the approximation becomes a nearest-neighbour map.

For the discrete case, if we wish to obtain the posterior probability of the output given the input and the model of the data, we would use the LSE estimator. On the other hand, if we wish to obtain output estimates that fall in our discrete output space we would use the STOCH estimator. [2]

Returning to the Gaussian mixture case, the STOCH and the SLSE estimators are more appropriate for learning non-convex inverse maps, where the mean of several solutions to an inverse might not be a solution. Both STOCH and SLSE take advantage of the explicit representation of the input/output density by selecting one of the several solutions to the inverse.

In the next three sections we illustrate, through case studies, the general phenomenon of non-convex inverses in learning. We also provide empirical evidence for the claim that density estimation using the STOCH or SLSE estimators, but not the LSE estimator, is a feasible approach to learning in these contexts.

Inverse kinematics

As a first example of a non-convex inverse problem we present the inverse kinematics of a three-joint planar arm. This problem involves learning the mapping between end-point Cartesian positions $\mathbf{x} = (x, y)$ and joint angles $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ of a robotic arm. Whereas the forward kinematic map from joint angles to end-point positions is always well-posed, it has been noted that redundancy of of the arm allows for many solutions to the inverse, causing a form of ill-posedness known as the "degrees-of-freedom problem" [1]. Approaches to learning the inverse kinematic map by sampling the $(\mathbf{x}, \boldsymbol{\theta})$ space and directly estimating a function $\boldsymbol{\theta} = \hat{f}(\mathbf{x})$ have met with some success [13]. However, as Jordan and Rumelhart (1992) have pointed out, the non-convexity of the map places a lower bound on the achievable error of any direct least-squares algorithm. Jordan and Rumelhart propose an indirect approach to this non-convexity problem based on forming an internal model of the arm and using this model to transform errors in Cartesian space to errors in joint-angle space.

Here we propose an alternative direct method where we will use our density estimation technique to form a model of the arm. Conditionalizing this density at values along the joint-angle space gives us a forward kinematic model of the arm. Conditionalizing at values along the end-point space gives us the inverse kinematic map. Since non-convexity implies that this latter conditional density is multimodal, we expect the LSE estimator to be inferior to the STOCH or SLSE estimators.

Figure 1 shows the results of learning the kinematics of an unconstrained three-joint planar arm with relative link lengths 1.0, 1.0, and 0.5. In Figure 1 (a) we see large reaching errors obtained on a feedforward backpropagation network using a least-squares error criterion to learn the inverse kinematics. Figure 1 (b) shows that first performing density estimation and then taking the conditional expectation (LSE) of the density yields qualitatively similar results to the last-squares backprop network. On the other hand, it can be seen in Figures 1 (c & d) that the density estimate actually contains enough information so that, if sampled properly with STOCH or SLSE, satisfactory solutions to the inverse can be obtained.

Acoustics of the vocal tract

The motor theory of speech perception proposes that knowledge of speech production is used in the perception of speech [14]. One form of the motor theory proposes that speech perception is the process of inverting an internal model of speech production. Thus, speech is perceived by taking a model of how phonemes arise from the vocal tract and predicting from the acoustic signal what the speaker's intended vocal tract configuration was—i.e. speech perception is an inverse acoustics problem.

[2] Here an analogy can be made to Boltzmann machine learning [9]. Boltzmann machines minimize the relative entropy between their state distribution and the target state distribution. This corresponds to maximum likelihood density estimation, taking the target distribution to be the empirical distribution of the data. Analogously, the EM approach to Bernoulli mixtures estimates the target density by placing the component means in parts of the space with high data density. Using the approximation $\hat{\mathbf{x}}^{\circ}(\mathbf{x}^{\mathrm{i}}) = \mu_j^{\circ}$ where $j = \arg\max_k P(z_k|\mathbf{x}_i^{\mathrm{i}})$ emulates basins of attraction by completing patterns with the probabilistically nearest mean, with the number of such "basins" equal to the number of components in the mixture. Finally, it is worth noting that Boltzmann machine learning is also an instance of generalized EM [9].

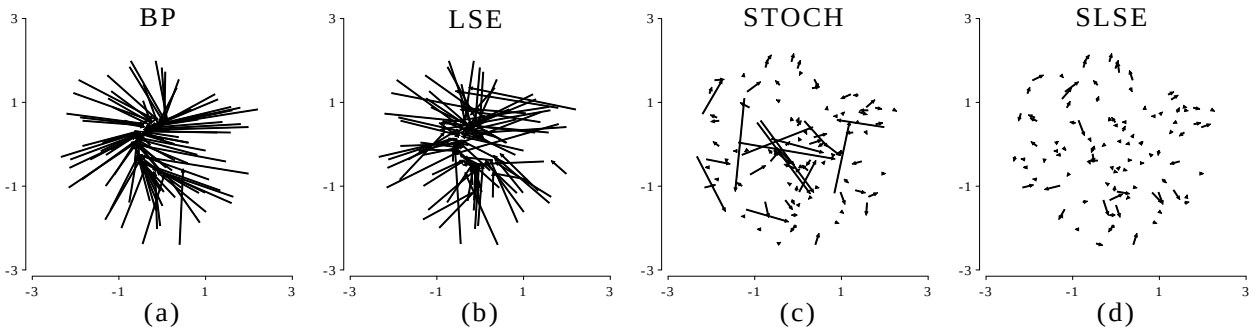

Figure 1. Learning direct inverse kinematics: Vector fields of reaching errors of a three-joint planar arm. Each of the above was trained on 1000 pairs of Cartesian (x, y) inputs and $(\theta_1, \theta_2, \theta_3)$ joint angle targets. The vectors are calculated on a test set of 200 points as the difference between an (x^*, y^*) command given to the network and the forward kinematic transformation of the output of the network $(x, y) = \mathrm{KIN}(\theta_1, \theta_2, \theta_3)$. (a) Backpropagation (RMS error = 1.211; a coarse search over the learning rate, momentum, and number of hidden units did not yield qualitatively different solutions from this). (b) EM Mixture of 60 Gaussians using the LSE estimator, (RMS error = 1.128). (c) Same mixture using the STOCH estimator (RMS error = 0.247). (d) Using the SLSE estimator (RMS error = 0.134).

In this section of the paper we observe that the non-convexity issue also arises in the context of this inverse acoustics problem. We limit our analysis to vowel production in a simplified four-tube model of the vocal tract [5] (see Figure 2 (a)). Tongue position and constriction of the model are varied as the tract resonances corresponding to the first three formants F_1,F_2 and F_3, are measured. These three formants are perceptually salient features of vowels in human speech. The learner is presumed to randomly sample the configuration space of the vocal tract, observe the vowels produced, and attempt to learn the mapping between vowels formants and tract configuration – essentially a static inverse acoustics problem. More sophisticated schemes involve a learner that uses a dynamic model of the articulators to recursively estimate the vocal tract configuration [10]. We will focus on the simpler static case.

One thousand data points were generated by randomly varying the tongue position between -6.5cm and 6.5cm and the tongue constriction between -1.0cm and 1.0cm about their resting states, and measuring the first three vowel formants (in Hertz). The learner estimated this 5 dimensional density using 60 full covariance Gaussians and 20 iterations of the EM algorithm, enough for approximate convergence. The density was then used to estimate tongue position and constriction $(\hat{x}_1, \hat{x}_2)$ from the formants. The acoustic outcome of this estimate was then compared to the actual input formants to obtain an error measure. Figures 2 (b& c) show that the least-squares estimates of tongue position and constriction obtained by taking the conditional expectation of the density do not correspond to the actual formants. On the other hand we see in Figure 2 (d & e) that the estimates obtained from the SLSE estimator can accurately reproduce the formants. The mean euclidean errors were 169.9 ± 5.6 Hz for the LSE estimator and 15.6 ± 1.5 Hz for the SLSE estimator (n=5 runs).

Thus, as with the inverse kinematics problem, non-convexity is of high relevance in predicting articulator configuration from formants, even though strictly speaking the problem in this case is not due to excess degrees of freedom but to symmetries in the vocal tract.

Localization of multiple objects from sensor data

As a final example of a non-convex learning problem we present the localization of multiple objects from sensor readings. The framework for this problem is one in which the learner is presented sensor readings from a room and the location of a *single* object in that room. The goal is to learn to determine from a sensor reading the locations of all the objects in the room. Given that there may be more than one object in the room contributing to the sensor readings at any time, we view this problem as one in which there are hidden sources and the learner is given incomplete data. The non-convexity in the problem arises from the hidden

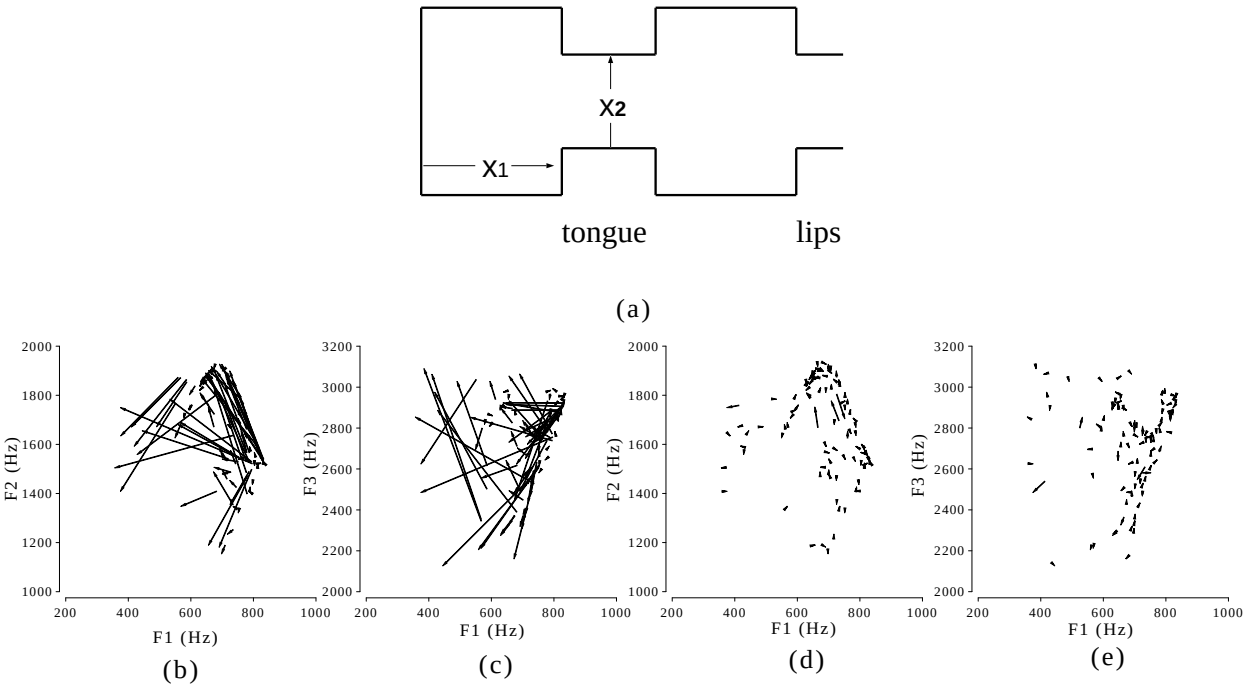

Figure 2. Four-tube model of the vocal tract. (a) Two parameters of the model, the tongue position x_1 and the tongue constriction x_2, are varied and the acoustic resonances of the four-tube model are computed. (b & c) Projections of the error vector field in formant space for the LSE estimator. (d & e) Projections of the error vector field for the SLSE estimator.

source: one object at the average location of two objects in the environment does not give the same sensor readings as the two objects combined.

For this particular example we assume that the contribution of each object in the room to the sensor readings is additive, and that a reading is inversely proportional to the square distance between the object and the sensor – a situation roughly analogous to point light sources being detected by light meters in a non-reflecting room. 1225 data points were generated by independently placing two objects, (x_1, y_1) and (x_2, y_2), on a grid in the room and calculating the sensor readings (s_1, s_2, s_3, s_4). The learner was trained using only the first object (i.e. on data points $(x_1, y_1, s_1, s_2, s_3, s_4)$) with a mixture of 60 Gaussians for 20 iterations of the EM algorithm.

Figure 3 shows four examples of the learner's estimated conditional density of object location given sensor readings calculated over the room, $\hat{P}(x, y|s_1, s_2, s_3, s_4)$. Four pairs of object locations were randomly generated and their corresponding sensor representations were computed as the input to the network. The density estimated by the learner is multimodal and tends to agree with the actual object locations; whereas a learner which simply attempts to predict (x, y) location by non-linear regression on the same data set would always falsely detect a single object intermediate between the two objects.

Two things should be noted about this example. First, even this simple multiple object localization problem suffers from exponential growth in number of data points as the number of objects increases. That is, for n objects each represented at a resolution of $1/k$ in each dimension, there are k^{2n} configurations of the room and sensor readings. This makes training the density estimator infeasible as the number of objects increases.

Second, it should be noted that for this example the generative model assumed by the mixture density does not reflect the way the data were actually generated. That is, even though the sensor data are the result of several simultaneous objects in the room, the mixture model assumes that each data point was generated by exactly one Gaussian. Thus, if the network output is a bimodal conditional density it should

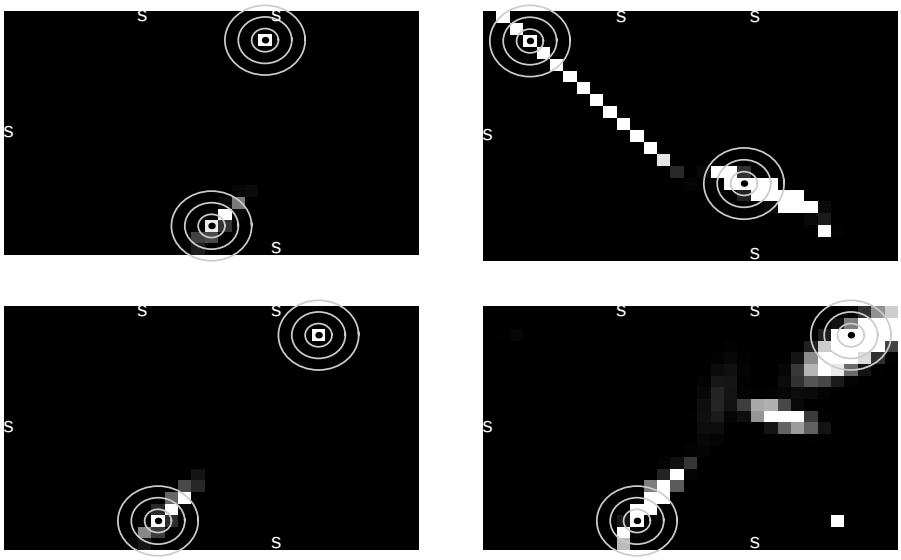

Figure 3. Localizing multiple objects from sensor data; four examples. The shading of the background represents the estimated probability density over object location given the sensor readings. (Shade at (x, y) is proportional to $\log(1+\hat{P}(x, y|s_1, s_2, s_3, s_4))$, with brighter representing higher estimated probability). The concentric circles mark at their center the actual locations of the two objects that generated the sensor data. The letter "s" marks sensor position. Note that the learner always estimated the actual locations to have relatively high probability (white squares).

be interpreted as a single object whose location is uncertain, not as two objects. However, from the data vectors alone there is no way of distinguishing between one object which gives the same sensor readings at two locations, and two objects, only one of which is present in the data vector.

An idea which may overcome the exponential growth of samples by explicitly representing the multiple causes in this data is the cooperative vector quantizer (CVQ) based on Minimum Description Length principles [8]. The CVQ extracts a compact code for the data by assuming that several independent sources collaborate to generate the data vector. In this example, the CVQ would extract such a code for the sensor readings, and the mapping from this code to sets of (x, y) positions could simultaneously be learned in a supervised fashion.

Discussion

Many learning problems do not fall under the rubric of traditional function approximation or classification. In this paper we have outlined one such class of problems, those involving non-convex learning, and an approach to solving them through parametric density estimation.

The three examples presented are instances of different sources of non-convexity. In the inverse kinematics problem the non-convexity arises from the excess degrees of freedom in a three-joint planar arm. In the vocal tract configuration problem the non-convexity arises from symmetries in the vocal tract. In the object localization problem it arises from the fact that the learner is presented with incomplete information about the environment—that is, it learns with one target object location at a time when there are in fact multiple objects in the room.

The particular density estimation procedure, applying maximum likelihood to a parametric mixture model using the EM algorithm, has the attractive properties that it generalizes to real and discrete data, can handle arbitrary patterns of incompleteness, and takes advantage of the convergence speed of EM. For

applicability to large problems full data-parallel implementations of this algorithm have also been coded on a Connection Machine CM5.

Further directions of research include extending the implementations, running on high dimensional real-world data sets, and testing an on-line weighted recursive least squares update rule.

References

[1] N. Bernstein, *The coordination and regulation of movements.* London: Pergamon, 1967.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees.* Belmont, CA: Wadsworth International Group, 1984.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Society Series B*, vol. 39, pp. 1–38, 1977.

[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.

[5] G. Fant, *Acoustic Theory of Speech Production.* Mouton and Co., 1960.

[6] J. H. Friedman, "Multivariate adaptive regression splines," *The Annals of Statistics*, vol. 19, pp. 1–141, 1991.

[7] Z. Ghahramani and M. I. Jordan, "Function approximation via density estimation using the EM approach," Computational Cognitive Science TR 9304, MIT, 1993.

[8] G. E. Hinton, "Using the minimum description length principle to discover factorial codes." Lecture given at the 1993 Connectionist Models Summer School, 1993.

[9] G. E. Hinton and T. J. Sejnowski, "Learning and relearning in Boltzmann machines," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations* (D. E. Rumelhart and J. L. McClelland, eds.), Cambridge, MA: MIT Press, 1986.

[10] J. F. Houde, "Recursive estimation of articulatory control," Computational Cognitive Science TR, MIT, Cambridge, MA, 1991.

[11] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," tech. rep., MIT, 1993.

[12] M. I. Jordan and D. E. Rumelhart, "Forward models: Supervised learning with a distal teacher," *Cognitive Science*, vol. 16, pp. 307–354, 1992.

[13] M. Kuperstein, "Neural model of adaptive hand-eye coordination for single postures," *Science*, vol. 239, pp. 1308–1311, 1988.

[14] A. M. Liberman, F. S. Cooper, D. P. Shankweiler, and M. Studdert-Kennedy, "Perception of the speech code," *Psychological Review*, vol. 74, pp. 431–461, 1967.

[15] S. J. Nowlan, *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, April 14 1991.

[16] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization.* New York: Wiley, 1992.

[17] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Networks*, vol. 2, no. 6, pp. 568–576, November 1991.