Institute for Neural Computation

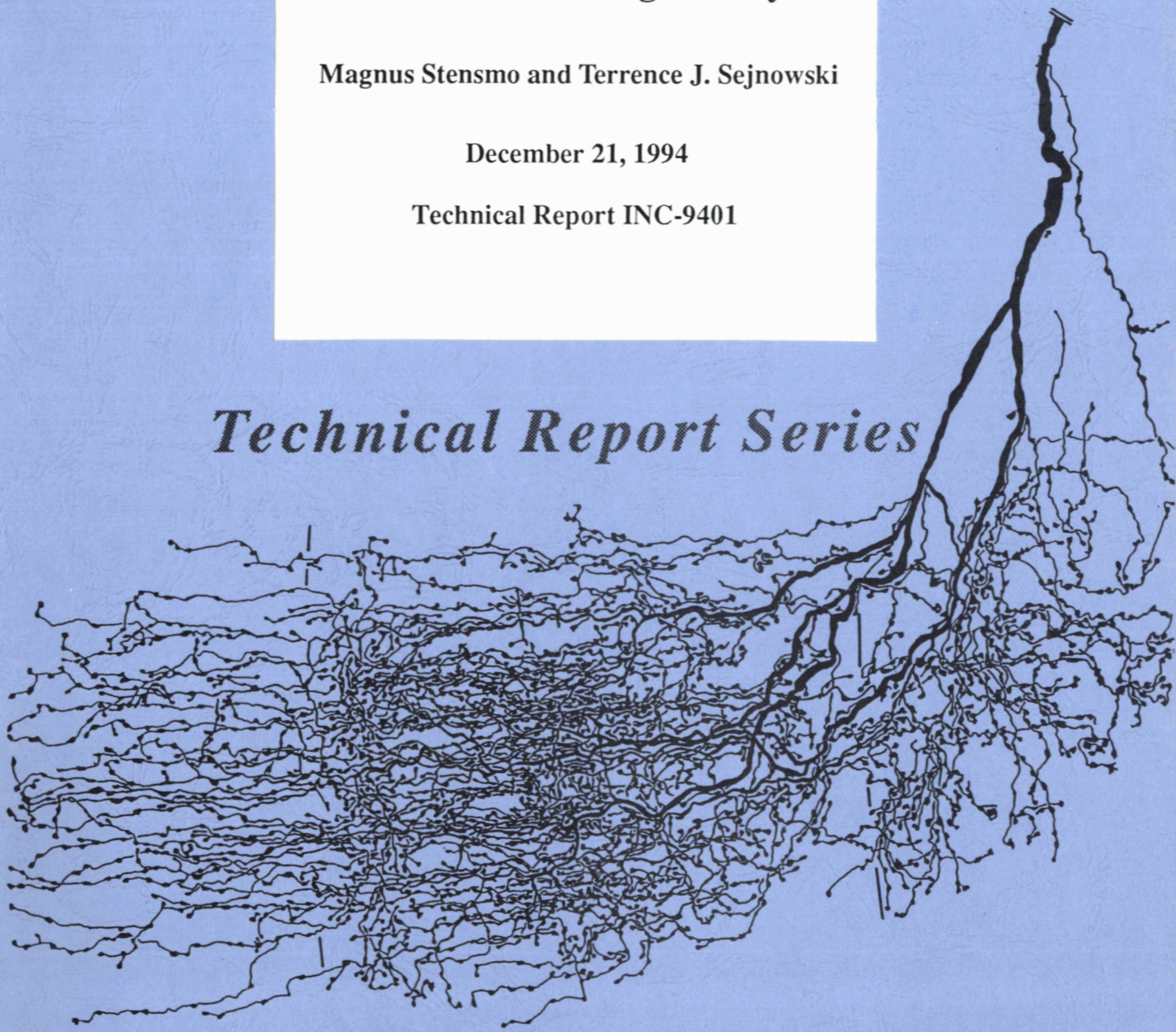
A Mixture Model Diagnosis System

Magnus Stensmo and Terrence J. Sejnowski

December 21, 1994

Technical Report INC-9401

Technical Report Series



University of California, San Diego La Jolla, California 92093

A Mixture Model Diagnosis System

Magnus Stensmo and Terrence J. Sejnowski

December 21, 1994

Technical Report INC-9401

Institute for Neural Computation
University of California, San Diego
9500 Gilman Drive, DEPT 0523
La Jolla, CA 92093-0523

Magnus Stensmo is with The Salk Institute and Terrence J. Sejnowski is with The Salk Institute and the Institute for Neural Computation at the University of California, San Diego.
Correspondence concerning this article may be addressed to: Magnus Stensmo, Computational Neurobiology Laboratory, The Salk Institute, 10010 North Torrey Pines Road, La Jolla, CA 92037.

A Mixture Model Diagnosis System

Magnus Stensmo* Terrence J. Sejnowski*†

*Computational Neurobiology Laboratory
The Salk Institute
10010 North Torrey Pines Road
La Jolla, CA 92037, U.S.A.
{magnus,terry}@salk.edu

†Institute for Neural Computation
University of California, San Diego
9500 Gilman Drive, DEPT 0523
La Jolla, CA 92093-0523, U.S.A.

Technical Report INC-9401

Abstract

Diagnosis is the process of identifying the disorders of a machine or a patient by considering its history, symptoms and other signs. Starting from possible initial information, new information is requested in a sequential manner and the diagnosis is made more precise. It is thus a missing data problem since not everything is known. We model the joint probability distribution of the data from a case database with mixture models. Model parameters are estimated by the EM algorithm which gives the additional benefit that missing data in the database itself can also be handled correctly. Request of new information to refine the diagnosis is performed using the maximum utility principle from decision theory. Since the system is based on machine learning it is domain independent. An example using a heart disease database is presented.

1 Introduction

1.1 Diagnosis

Diagnosis is the process of identifying a disease or disorder of a patient or a machine by considering its history, symptoms and other signs. This is done by examination in various ways. The term also refers to the problem identification by the examiner. It is a common and important problem that is performed daily by many people in different professions.

An essential part of the diagnostic process is the reasoning. False diagnosis often result from wrong interpretations of findings:

> While the task of collecting data on the patient-history, physical examination, special examinations, and laboratory tests—requires accurate observation, the process of correct reasoning is, in the last analysis, more important. False diagnoses often result not from false data but from faulty interpretation. To assemble the data, to make use of the relevant and to discard the irrelevant information, and to pass final judgment—all of which are important steps in making the diagnosis—often requires a mental ability of the highest order.
>
> *Diagnosis* [Encyclopædia Britannica, 1992]

Diagnosis has proven hard to automate and formalize because the experts (physicians, engineers, *etc.*) themselves don't know exactly how they solve a problem. They acquire their skill with both study and experience. The objects they examine can also be very complicated and not fully understood—as in the case of the human body—but even a man-made machine can behave in unexpected ways. For these reasons, a procedural description might be hard or impossible to attain. It is hard to write a program that performs diagnosis at human levels.

In this paper we use the information about a specific problem that exists in a database of cases for the domain. The underlying relationships between the variables may be obscure, but they are nevertheless there provided the database is sufficiently complete. By considering this as a statistical inference problem we can attempt to understand the underlying relationships. We will use machine learning with our previous observations as a knowledge base.

Viewing diagnosis as a probability estimation problem, we imagine a set of possible disorders, and a set of variables that describe the situation. In a medical context the disorders are diseases and the variables are any data about the patient that we need for the diagnosis. The goal is to find the probability distribution over the disorders, or more precisely, conditional probabilities that are conditioned on what has been observed. The diagnosis is strong when one or a few of all of the possible outcomes are differentiated from the others. If it is inconclusive, so that several of the outcomes are more or less equally likely, more information is needed.

Machine diagnosis and medical diagnosis follow the same steps even though the domains, variables and disorders are different. When disease and symptom is mentioned below, disorder and observation can be equivalently substituted.

Diagnosing something can be viewed as solving a missing data problem. Initially we might have a few clues, but the rest of the variables are essentially unknown except for *a priori* probabilities. If our initial knowledge is not sufficient more information is needed. We get this by asking a question, the answer to which we get by performing some kind of test. Since tests may be both expensive and time consuming it is generally not possible or desirable to find the answer to every question. In any case we want to avoid unnecessary tests.

In a medical situation tests may be unpleasant, painful or even dangerous to the patient and should thus be avoided when possible. Irrelevant tests also take time which may be both critical to the health of the patient and costly. In a machine diagnosis context, there are generally varying costs associated with the tests, and the total cost should be minimized. This is why diagnosis is a process. There must thus be an efficient way to acquire new information dynamically. Starting from a few prior observations the diagnosis is vague. As more information is gathered the diagnosis becomes more precise.

1.2 Automated diagnosis

Due to the commonality of the diagnosis problem there have been many attempts to automate it, summarized in Table 1. There are problems with all of them which is what motivated the present work.

Early work used Bayesian reasoning and decision theory [Ledley & Lusted, 1959] and realized that the problem is not always tractable due to the large number of influences there can exist between combinations of symptoms and diseases. They even built a mechanical "learning device" that used cards with notches to keep track of patient cases that were similar.

Diagnosis systems have been successfully built using expert systems, *e. g.* the INTERNIST system for internal medicine [Miller *et al.*, 1982], later renamed QMR for Quick Medical Reference. MYCIN for blood infections [Shortliffe, 1976] is another example. Rule-bases are, however, very hard and time consuming to build and inconsistencies may arise when new rules are added to an existing database. There is also a strong domain dependence and knowledge bases can rarely be reused for new applications.

Approaches to diagnosis that are based on domain-independent machine learning alleviate some of the problems with knowledge engineering. Decision trees [Quinlan, 1986] have also been used. They too have problems with incomplete data, and a piece of information can only be used if the appropriate question comes up when traversing the tree. Irrelevant questions can not be avoided since the tree is always traversed in the same way starting from the root. A decision tree is a static representation.

Multi-layer perceptrons have been successfully applied to diagnosis problems.

Model	Specification	Missing data	Domain specific	Limitations
Expert system	manual	no	yes	Hard to build, Rule-base inconsistencies
Decision tree	learning	no	no	Missing or initial data in traversal order
Feed-forward neural network	learning	no	no	All inputs must be known
Bayesian network	manual	yes	yes	Probabilities must be estimated
"Simple Bayes" network	learning	yes	no	Independence assumptions not valid
Mixture model system	learning	yes	no	Large data sets needed

Table 1: Problems with previous approaches to the diagnosis problem. See text in Section 1.2.

In fact, most feed-forward systems estimate posterior probabilities with a suitable encoding of the output units [Richard & Lippmann, 1991]. Feed-forward multi-layer perceptrons for diagnosis [Baxt, 1990; Shavlik *et al.*, 1991] can classify very well, but they need full information about a case, *i. e.* the values of all of the variables at once. This is something that almost never will happen in a real situation. Moreover, they cannot handle missing data, neither during learning nor during the classification. Success or failure often depends on the input representation. This can make the approach more or less domain independent.

The large numbers of probabilities involved can make exact diagnosis intractable. With n variables there are $2^n - 1$ combinations of the variables. Approximations are necessary. Probability estimation in a simplified context has been investigated by [Gorry & Barnett, 1967; Kononenko, 1989; Stensmo, 1991]. A simple way to obtain an approximation is to assume independence between all variables and conditional independence given the disease. There is then a simple equation for the conditional probabilities.

These assumptions are of course not true since there clearly are dependencies between the symptoms and diseases. Because of this fact, the systems are called *simple*, *naive* or *idiot's Bayes*. Although they work surprisingly well in many instancies, this is not guaranteed. An equivalent one-layer (*i. e.*, linear) neural network can be formulated [Stensmo, 1991]. This demonstrates the weakness of the approximation.

Bayesian networks, [Pearl, 1988; Henrion *et al.*, 1991], is an interesting way to model a joint probability distribution by factoring using the chain rule in probability theory. Although the models are very powerful when built, there are presently no general machine learning methods for their construction. A considerable effort is needed, for example, in the Pathfinder system for lymph node pathology [Heckerman *et al.*, 1992; Heckerman & Nathwani, 1992], about 14000 conditional probabilities had to be assessed by an expert pathologist.

They had to divide the problem into independent partitions, called probabilistic similarity networks [Heckerman, 1991], to make it manageable. This allowed them to assess only a fraction of the 75000 original probabilities, but what remains is still of considerable size. It is inevitable that errors will occur when such large numbers of assessments are involved. An additional drawback is that general probabilistic inference on Bayesian networks is NP-hard, even for restricted networks.

In the present work, the probability distributions are approximated by a method called mixture models. The joint probability distribution is modeled so that the missing data problem is solved [Ahmad & Tresp, 1993].

2 Probability Estimation

Diagnosis is a probability estimation problem. The probabilities of the outcomes are conditioned on what has currently been observed. There are several ways to determine the conditional probability of a disorder C_i given a set of variables X,

$p(C_i|X)$.

- The *direct* approach is to learn a functional approximation from the inputs, X, to the outputs, C. These are thus the conditional probabilities $p(C_i|X)$ for each disorder i.
- An *indirect* way is to rewrite the conditional probability using Bayes's rule

$$p(C_i|X) = \frac{p(X|C_i)p(C_i)}{p(X)} \tag{1}$$

  and estimate the parts—especially the new conditional—separately.
- A third way is to model the *joint probability* $p(C, X)$. If this is known it is easy to marginalize to get any conditional probability. It is necessary to model the joint probability to be able to handle missing data in a principled way [Ahmad & Tresp, 1993]. We adopt this approach.

The first two can be modeled with feed-forward neural networks [Hush & Horne, 1993; Richard & Lippmann, 1991]. This will however not work for subsets of X since a neural network classifier needs all input values at the same time.

Classification with unknown inputs has recently been addressed in the neural network context [Ahmad & Tresp, 1993; Ghahramani & Jordan, 1994]. They used mixture models of normal distributions whereby simple closed form solutions to optimal regression with missing data can be formulated. The EM algorithm for parameter estimation is especially interesting in this context since it can also be formulated to handle missing data in the training examples [Dempster *et al.*, 1977; Ghahramani & Jordan, 1994; Tresp *et al.*, 1994]. Most real world data sets have missing data.

2.1 Mixture models and EM

A method from parametric statistics called *mixture models* [McLachlan & Basford, 1988] is good for modeling joint probability distributions. It has recently been used for supervised and unsupervised learning problems [Nowlan, 1991; Jacobs *et al.*, 1991; Ghahramani & Jordan, 1994]. A simple closed form solution for an optimal solution to the missing data problem [Ahmad & Tresp, 1993] can be achieved with appropriate mixture components. The method is parametric in the sense that the form of the mixture components have to be pre-defined.

The data underlying the model is assumed to be a set of N vectors $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, each of dimension D. Each data point $\boldsymbol{x}$ is assumed to have been generated independently from a *mixture density* with M components

$$p(\boldsymbol{x}) = \sum_{j=1}^{M} p(\boldsymbol{x}, \omega_j; \theta_j) = \sum_{j=1}^{M} p(\omega_j)p(\boldsymbol{x}|\omega_j; \theta_j), \tag{2}$$

where each mixture component is denoted by ω_j. $\mathrm{p}(\omega_j)$, the *a priori* probability for mixture ω_j, and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M)$ are parameters.

Our goal is to estimate the parameters for the different mixtures so that it is likely that the linear combination of them generated the set of data points. This is accomplished by *maximum likelihood estimation.*

The *Expectation-Maximization*, or *EM*, algorithm is an iterative maximum likelihood estimation technique that is simple and converges quite rapidly.

Two steps are repeated:

1. First a likelihood is formulated and its expectation is computed in the *Estimation* or *E-step.* For the type of models that we will use, this step will calculate the probability that a certain mixture component generated the data point in question.

2. The second step is the *Maximization* or *M-step.* The parameters that maximize this likelihood are found. This maximization can be found analytically for a certain set of models that can be written in an exponential form, *e. g.*, Gaussian functions.

The EM algorithm is quite sensitive to initial conditions. However, initial estimates obtained by the k-means algorithm [Duda & Hart, 1973; Tou & Gonzales, 1974] are sufficient. Equations can be derived for both batch and on-line learning.

Update equations for Gaussian and binomial distributions with and without missing data will be given here. Details and derivations can also be found in [Dempster *et al.*, 1977; Nowlan, 1991; Ghahramani & Jordan, 1994]. First, update equations for complete data are given, followed by the incomplete case.

2.2 EM for complete data

From (2) we form the log likelihood of the data

$$L(\boldsymbol{\theta}|\boldsymbol{X}) = \sum_{i=1}^{N} \log \mathrm{p}(\boldsymbol{x}_i; \theta_j) = \sum_{i=1}^{N} \log \sum_{j=1}^{M} \mathrm{p}(\omega_j)\mathrm{p}(\boldsymbol{x}_i|\omega_j; \theta_j) \tag{3}$$

There is unfortunately no analytic solution to the log of a sum in the right hand side of the equation. However, if we were to know which of the mixtures that generated which data point we could compute it. The EM algorithm solves this via a "trick": A new set of binary indicator variables $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$ is introduced. $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{iM})$, where $z_{ij} = 1$ if and only if the data point $\boldsymbol{x}_i$ was generated by mixture component j.

As shown in [Dempster *et al.*, 1977; McLachlan & Basford, 1988], the log likelihood can be manipulated to a form that does not contain the log of a sum

$$L(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Z}) = \sum_{i=1}^{N}\sum_{j=1}^{M} z_{ij} \log \mathrm{p}(\boldsymbol{x}_i, \boldsymbol{z}_i; \boldsymbol{\theta}) = \sum_{i=1}^{N}\sum_{j=1}^{M} z_{ij} \log \left(\mathrm{p}(\boldsymbol{z}_i; \boldsymbol{\theta})\mathrm{p}(\boldsymbol{x}_i|\boldsymbol{z}_i; \boldsymbol{\theta})\right). \tag{4}$$

z_i is not known which means that (4) cannot be used directly. Its expectation using the current parameter values $\boldsymbol{\theta}_k$ is instead used. This is the *E-step* of the EM algorithm

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}_k) = \mathrm{E}[L(\boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{Z})|\boldsymbol{X}, \boldsymbol{\theta}_k] \tag{5}$$

In the E-step, $Q(\boldsymbol{\theta}|\boldsymbol{\theta}_k)$ simplifies to $\mathrm{E}[z_{ij}|\boldsymbol{x}_i, \boldsymbol{\theta}_k]$, see [McLachlan & Basford, 1988].

When the expected value is known it can be maximized in the *M-step*

$$\boldsymbol{\theta}_{k+1} = \mathrm{argmax}_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}_k) \tag{6}$$

This can be done analytically for distributions that can be written in an exponential form (Generalized Linear Models [Dobson, 1990]).

The two steps are iterated until convergence. It can be shown that the likelihood will never decrease after an iteration [Dempster *et al.*, 1977]. It is often reported that convergence is slow; however, it is fast compared to gradient descent. For a longer discussion of this, see [Xu & Jordan, 1994].

2.3 EM for missing data

One of the main motivating ideas behind the EM-algorithm was to be able to handle missing values for variables in a data set in a principled way.

In the complete data case we introduced missing indicator variables that helped us solve the problem. With missing data we add the missing components to the $\boldsymbol{Z}$ that were missing before, and solve as above [Dempster *et al.*, 1977; Ghahramani & Jordan, 1994]. Each vector $\boldsymbol{x}_i$ is split up into $(\boldsymbol{x}_i^o, \boldsymbol{x}_i^m)$, where o denotes observed data and m missing data.

The M-step remains as in the complete data case. The E-step becomes

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}_k) = \mathrm{E}[L(\boldsymbol{\theta}|\boldsymbol{X}^o, \boldsymbol{X}^m, \boldsymbol{Z})|\boldsymbol{X}^o, \boldsymbol{\theta}_k]. \tag{7}$$

2.4 EM for normal distributions

Below are given the specialization of the EM algorithm to the case where the mixture components are Gaussian distributions. A Gaussian for mixture component j with mean $\boldsymbol{\mu}_j$ and a full covariance matrix Σ_j can be written

$$\mathrm{p}(\boldsymbol{x}|\omega_j) = G_j(\boldsymbol{x}) = \left(\frac{1}{\sqrt{2\pi}}\right)^D |\Sigma_j|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_j)\right]. \tag{8}$$

In the general case the covariance matrix is full, *i. e.* $D \times D$. Both storage and computational complexity then grows quite fast ($\mathcal{O}(D^2)$). The form of the covariance matrix is therefore often constrained to be diagonal or to have the same values on the diagonal, $\Sigma_j = \sigma_j^2 I$ so that the complexity becomes $\mathcal{O}(D)$. This corresponds to axis-parallel oval-shaped and radially symmetric Gaussians, respectively. Radial and

diagonal basis functions has functioned well in many applications [Nowlan, 1991], since several Gaussians together can form complex shapes in the space. With fewer parameters over-fitting is minimized.

In the radial case the Gaussians have the standard form

$$G_j(\boldsymbol{x}) = \left(\frac{1}{\sqrt{2\pi}\sigma_j}\right)^D \exp\left[-\frac{\| \boldsymbol{x} - \boldsymbol{\mu}_j \|^2}{2\sigma_j^2}\right], \tag{9}$$

with variance σ_j^2 and mean as above.

Complete data. In the E-step the expected value of the likelihood is computed. For the Gaussian case this becomes the probability that Gaussian j generated data point $\boldsymbol{x}$, or

$$p_j(\boldsymbol{x}) = \frac{\mathrm{p}(\omega_j)G_j(\boldsymbol{x})}{\sum_{k=1}^{M} \mathrm{p}(\omega_k)G_k(\boldsymbol{x})} \tag{10}$$

The M-step finds the parameters that maximize the likelihood from the E-step. We will re-estimate the *a priori* probability, the means and the widths of the Gaussians. For time step $t+1$ the update equations become

$$\hat{p}(\omega_j) \longleftarrow \frac{S_j}{N}, \tag{11}$$

$$\hat{\boldsymbol{\mu}}_j \longleftarrow \frac{1}{S_j}\sum_{i=1}^{N} p_j(\boldsymbol{x}_i)\boldsymbol{x}_i, \tag{12}$$

$$\hat{\sigma}_j^2 \longleftarrow \frac{1}{DS_j}\sum_{i=1}^{N} p_j(\boldsymbol{x}_i) \| \boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_j \|^2, \tag{13}$$

where

$$S_j = \sum_{i=1}^{N} p_j(\boldsymbol{x}_i). \tag{14}$$

Incomplete data. When input variables are missing the $G_j(\boldsymbol{x})$ in (10) is only evaluated over the set of observed dimensions O. Missing (unobserved) dimensions are similarly denoted U, so that $|D| = |O| + |U|$. (9) now becomes

$$G_j^O(\boldsymbol{x}) = \left(\frac{1}{\sqrt{2\pi}\sigma_j}\right)^{|O|} \exp\left[-\frac{\| \boldsymbol{x}^O - \boldsymbol{\mu}_j^O \|^2}{2\sigma_j^2}\right]. \tag{15}$$

The update equation for $\hat{p}(\omega_j)$ is unchanged. To estimate $\hat{\boldsymbol{\mu}}_j$ we set $\boldsymbol{x}_i^U = \hat{\boldsymbol{\mu}}_j^U$ and use (12). The variance becomes

$$\hat{\sigma}_j^2 \longleftarrow \frac{1}{DS_j}\sum_{i=1}^{N} p_j^O(\boldsymbol{x}_i)\left[\| \boldsymbol{x}_i^O - \hat{\boldsymbol{\mu}}_j^O \|^2 + |U|\hat{\sigma}_j^2\right]. \tag{16}$$

Regression. To fill in missing data values during classification a least squares regression is used. Missing dimensions are denoted U as above, and the completed data is

$$\hat{\boldsymbol{x}}^U = \sum_{j=1}^{M} p_j^O(\boldsymbol{x}^O)\boldsymbol{\mu}_j^U. \tag{17}$$

The regression is the conditional expectations given the observed variables. In the Gaussian case

$$\mathrm{E}[\boldsymbol{x}^{\mathrm{m}}|\boldsymbol{x}^{\mathrm{o}}] = \sum_{|O|} \boldsymbol{x}^{\mathrm{o}}\mathrm{p}(\boldsymbol{x}^{\mathrm{m}}|\boldsymbol{x}^{\mathrm{o}}) \tag{18}$$

With missing variables (and radial Gaussians) this is the same as projection the onto the known dimensions [Ahmad & Tresp, 1993] followed by evaluation.

Classification. The result of the regression when the disorder variables are missing is a probability distribution over the disorders. This can be reduced to a classification for comparison with other systems by picking the disorder with the maximum of the estimated probabilities.

2.5 EM for mixtures of binary variables

Assume that each vector $\boldsymbol{x} = (x_1, \ldots, x_D)$ is a D dimensional binary vector. For each mixture component ω_j there is a parameter vector μ_j.

$$\mathrm{p}(\boldsymbol{x}|\omega_j) = \prod_{d=1}^{D} \mu_{jd}^{x_d}(1-\mu_{jd})^{(1-x_d)} \tag{19}$$

This means that the factor μ_{jd} is used if $x_d = 1$ and $1 - \mu_{jd}$ otherwise. This is clearer in the log version:

$$\log \mathrm{p}(\boldsymbol{x}|\omega_j) = \sum_{d=1}^{D}[x_d \log \mu_{jd} + (1-x_d)\log(1-\mu_{jd})] \tag{20}$$

The E-step is

$$p_j(\boldsymbol{x}_i) = \frac{\mathrm{p}(\omega_j)\mathrm{p}(\boldsymbol{x}_i|\omega_j)}{\sum_{k=1}^{M}\mathrm{p}(\omega_k)\mathrm{p}(\boldsymbol{x}_k|\omega_j)} \tag{21}$$

In the M-step we estimate the vector $\boldsymbol{\mu}_j$

$$\hat{\boldsymbol{\mu}}_j \longleftarrow \frac{\sum_{i=1}^{N} p_j(\boldsymbol{x}_i)\boldsymbol{x}_i}{\sum_{i=1}^{N} p_j(\boldsymbol{x}_i)} \tag{22}$$

Regression and classification is as in the Gaussian case.

3 Requesting more information

During the diagnosis process, the result is refined in each step based on newly acquired knowledge. It it important to select the right questions to ask and to perform the minimal number of tests necessary. There is generally a cost associated with each possible test. The goal is to minimize the cost while requesting as few tests as possible. In a medical context we might not of ethical reasons want to use costs if the object is to make the patient well. Instead we weight in the implications for the patient in terms of inconvenience, pain or risk of future complications.

Early work on automated diagnosis [Ledley & Lusted, 1959] acknowledged the problem of asking as few questions as possible and suggested the use of decision analysis for the solution. Decision theory is derived from the theory of games by [von Neuman & Morgenstern, 1947].

A very important idea from the field of decision theory is the *maximum expected utility principle* [von Neuman & Morgenstern, 1947]. The idea is that a decision maker should always choose the alternative that maximizes some expected utility of the decision. It has been thought that we normally behave according to this principle.

In the current context we are interested in the *utility of misclassification.* Each pair of outcomes has a utility $u(x, y)$ when the correct diagnosis is x but y has been incorrectly determined.

The utility values have to be assessed manually in what can be a lengthy and complicated process. For n disorders there are $n(n-1)/2$ combinations, and it thus grows $\mathcal{O}(n^2)$. For this reason a simplification of this function has been suggested by [Heckerman *et al.*, 1992]: The utility $u(x, y)$ is 1 when both x and y are benign or both are malign, and 0 otherwise. This simplification has been found to work well in practice. The amount of utility assessments has thereby been greatly simplified.

There is another complication with the maximum expected utility principle that can make it intractable. In the ideal case we would evaluate every possible sequence of future choices to see which would be the best. Since the size of the search tree of possibilities grows exponentially this is of course not reasonable. A simplification [Gorry & Barnett, 1967] is to only look ahead one or a few steps at a time. This nearsighted or *myopic* approach has been tested in practice with good results [Gorry & Barnett, 1967; Heckerman *et al.*, 1992].

In the last section another possible method for requesting more information is discussed.

4 The Diagnosis System

The system has two phases. First there is a learning phase where a probabilistic model is built. The model is then used for inference in the diagnosis phase.

4.1 Learning phase

The joint probability distribution of the data is modeled as described in Section 2 using mixture models. Parameters are determined by the EM algorithm. The k-means algorithm is used for initialization. These steps make up the learning phase where the model is built.

Variable and disorder variables for each case are combined into one vector per case and form the set of training patterns. Since there are no differences between input and output in the mixture model we can use it to do both diagnosis and question selection.

Nominal variables, like the disorders, are coded as 1 of N. Continuous variables are interval coded. For the experiments in this paper, the coding is based on how many standard deviations the value is away from its mean. In this new coding the variables represent probabilities of occurrence within the different ranges.

4.2 Diagnosis phase

A sequential diagnosis is performed in the diagnosis phase. It consists of the following steps and uses the model that was built in the learning phase. Figure 1 shows a flowchart of this phase.

1. Any initial observations can be entered. Note that this really means any observations. In comparison, in a decision tree only some initial data can be used, depending on whether it occurs close to the root of the tree or not. It is similar for rule-based systems.

2. Regression is used to find expected values of unknown variables. The unknown dimensions are filled in using

$$\hat{\boldsymbol{x}}^U = \sum_{j=1}^{M} p_j^O(\boldsymbol{x}^O)\boldsymbol{\mu}_j^U. \tag{23}$$

   This is a rigorous way of handling missing data. A conditional expectation $E[\boldsymbol{x}^U|\boldsymbol{x}^O]$ is found instead of an unconditional one $E[\boldsymbol{x}^U]$, the *a priori* distribution.

3. The Maximum Expected Utility Principle is used to recommend the next observation to make.

   Subtract the cost of the observation if applicable.

   Stop if it is determined that nothing would be gained by making another observation. This happens when the expected utility is zero or negative.

4. The user is asked to determine the correct value for the recommended observation.

Table 2: The variables and disorders for the Cleveland Heart Disease database that is used in Section 5.

	Observation	Description	Values
1	age	Age in years	continuous
2	sex	Sex of subject	male/female
3	cp	Chest pain	four types
4	trestbps	Resting blood pressure	continuous
5	chol	Serum cholesterol	continuous
6	fbs	Fasting blood sugar	<, or > 120 mg/dl
7	restecg	Resting electrocardiographic result	five values
8	thalach	Maximum heart rate achieved	continuous
9	exang	Exercise induced angina	yes/no
10	oldpeak	ST depression induced by exercise relative to rest	continuous
11	slope	Slope of peak exercise ST segment	up/flat/down
12	ca	Number major vessels colored by flouroscopy	0-3
13	thal	Defect type	normal/fixed/ reversible
	Disorder	**Description**	**Values**
14	num	Heart disease	Not present/ four types

The user can always choose to do any other observation instead of or in addition to the recommendation.

5. Continue with step 2.

5 Example

The Cleveland heart disease data set from University of California, Irvine [Murphy & Aha] has been used to test the system. It contains 303 examples of four types of heart disease and its absence. There are thirteen continuous or nominally valued variables, see Table 2.

The continuous variables were interval coded with one unit for each standard deviation away from the mean value. This was chosen since they were approximately normally distributed. Nominal variables were coded with one unit per value. In total the 14 variables were coded with 55 units.

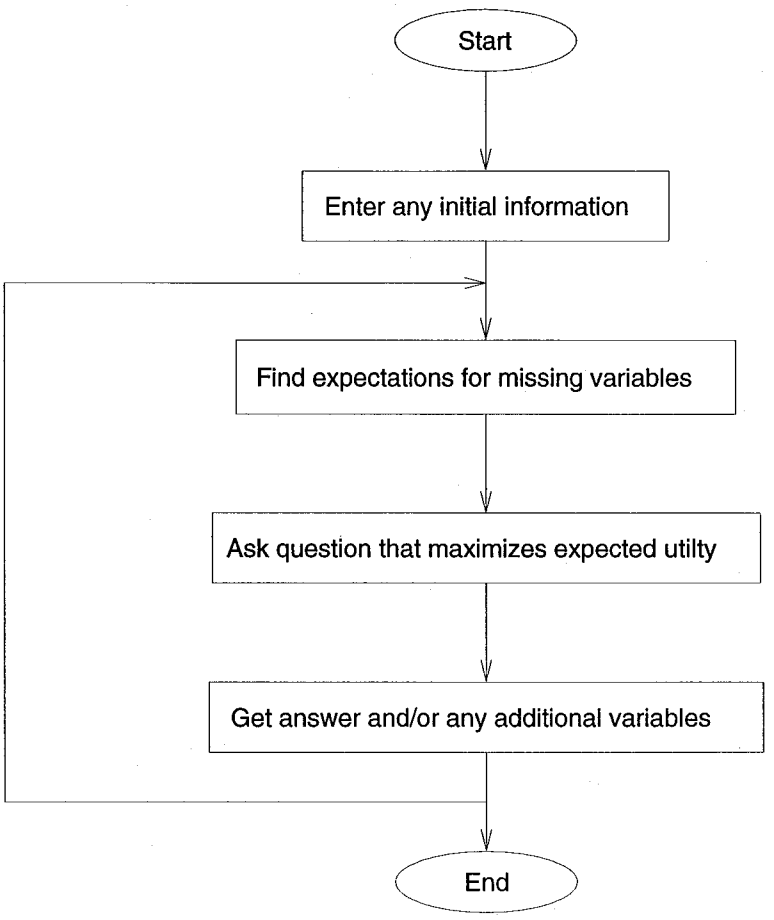


Figure 1: Flowchart of the Neural Expert System
This flowchart shows the basic operation of the system. When the value of a variable is requested (*i. e.*, the answer to a question), any number of questions can be answered. This sequence of questions and answers are repeated until there is a satisfactory separation between the output disorders. See Section 4.2.

Note that no background knowledge about the domain have been supplied and this diagnosis system is thus, in contrast to expert systems, domain independent. The EM algorithm is sensitive to initial conditions and starting values for the means and widths of the Gaussians were obtained by the k-means algorithm as mentioned above. Thereafter the EM steps were repeated until convergence, which took about 60–150 iterations of the two steps. A varying number of mixture components (20-120) were used.

Previously reported results have used only presence or absence of the heart disease. The best of these has been a classification rate of 78.9% with a system that incrementally builds prototypes [Gennari *et al.*, 1989]. We have obtained 78.6% with 60 radial Gaussian mixtures as described above. This shows that radial Gaussian mixture models can form good classifiers. A search for an optimal number of mixture components has not been done because of the known low quality of the data set [Gennari *et al.*, 1989], but performance increases as the number of mixture components go up. It also does not seem to be very sensitive to a varying number of mixture components unless there are too few of them.

Using all five disorders, instead of only presence and absence, only 66.6% correctly classified patterns could be achieved. Note that previous investigators of this data set have pointed out that there is not enough information in the thirteen variables to completely classify the disorders [Gennari *et al.*, 1989].

To show how the complete system works in practice an annotated transcript is shown in Figure 2. In this example, all of the variables are initially unknown, and their unconditional prior probabilities are used as starting values for the diagnosis.

6 Summary and Conclusions

We have explored diagnosis in the framework of probability estimation with missing data. The parameters of a mixture model of Gaussian basis functions were adjusted using the EM algorithm. It was shown how questions can be selected automatically. A system was built that used any possible initial information and gave estimates of probabilities for the different diagnosis disorders. To refine the diagnosis, more information is requested about the variable that is expected to be most useful following the maximum expected utility principle. This cycle of questions and answers is repeated until a satisfactory conclusion is reached.

A natural question at this point may be why we bother to build an automated diagnosis system when we already have doctors and specialists for every field of study. The answer is that it is a resource allocation problem. In the western world we certainly have enough physicians, but in many other areas there are too few of them. A system for automatic diagnosis could be helpful in such contexts. It also takes time and costs money to train technicians for machine diagnosis.

Initially all of the variables are unknown and the initial estimates for the five outcomes are the unconditional prior probabilities. The leftmost number of the five numbers in a line is the estimated probability for no heart disease, followed by the probabilities for the four types of heart disease given in the data set.
The entropy, defined as $-\sum_i p_i \log p_i$, of the diagnoses are given at the same time as an indication of how decisive the current conclusion is. A completely determined diagnosis—with one 1 and the rest 0—has entropy value 0.

```
Disorders (entropy = 1.85)
  0.541254  0.181518  0.118812  0.115512  0.042904
What is cp ? 3
```

The first question is "chest pain", and the answer changes the estimated probabilities. This variable is continuous. The answer is to be interpreted how far from the mean the observation is in standard deviations.

```
Disorders (entropy = 0.69)
  0.888209  0.060963  0.017322  0.021657  0.011848
What is age ? 0
```

Note that as the decision becomes more conclusive, the entropy decreases.

```
Disorders (entropy = 0.57)
  0.91307619  0.00081289  0.02495360  0.03832095  0.02283637
What is oldpeak ? -2

Disorders (entropy = 0.38)
  0.94438718  0.00089016  0.02539957  0.02691099  0.00241210
What is chol ?  -1

Disorders (entropy = 0.11)
  0.98848758  0.00028553  0.00321580  0.00507073  0.00294036
```

We have now determined that the probability of no heart disease in this case is 98.8%. The remaining 0.2% is spread out over the other possibilities.

Figure 2: Diagnosis example. The estimated probabilities for the different target disorders given the current observations are output and a new question is generated depending on the answers received. The variables are described in Table 2. See text in Section 5.

6.1 Further work

Several properties of this model remain to be investigated:

- It has to be tested on several more databases. Unfortunately it is very hard to find databases since they are mostly proprietary. Future prospects for medical databases should be good. There are now hospitals that have totally abandoned paper patient journals and are instead using computerized systems. Insurance companies and HMOs may find it in their interest to develop systems of the proposed type. It should be fairly easy to generate data for machine diagnosis.

- An alternative way to choose a new question is to evaluate what the variance change in the output variables will be when a variable is changed from missing to observed. The idea is that a variable known with certainty has zero variance [Chávez & Henrion, 1994]. Therefore, the variable with the largest *conditional variance* is selected as the query. A similar technique is the concept of *Optimal Experiment Design* [Fedorov, 1972], where experiments are designed to minimize the variance of a parameterized model, thereby maximizing the confidence in the model. This has been introduced into the neural network domain for other purposes [Cohn, 1994; MacKay, 1992].

- The robustness when questions are answered wrongly should also be tested. This has earlier been investigated in a limited "naive" Bayes context [Stensmo, 1991] with good results. In that system, the given answers and the system's own estimates were compared. If they differed, the questions might (inadvertently) have been incorrectly answered and are asked again for confirmation. Quantitative analysis of the number of questions needed to reach a conclusion compared to a random or worst case also remains to be investigated.

- One important aspect of a diagnosis system is its ability to explain why it reached a certain conclusion. This factor is important for user acceptance of an automated diagnosis system. Since the basis functions have local support and since we have estimates of the probability of each function having generated the observed data, the explanation for the conclusion can be found. This has been used by [Tresp *et al.*, 1994] in a different but related context.

- Instead of using the simplified utilities with values 0 and 1 for the expected utility calculations it would be interesting to investigate the learning of utilities using reinforcement learning. A trained expert would evaluate the quality of the diagnosis performed by the system, followed by adjustment of the utilities so that a good result is rewarded and a bad result is punished. Even though a human expert is needed, the work needed is minor compared to the assessment of all of the utility pairs. The 0 and 1 values can be used as starting values.

Acknowledgements

The heart disease database is from the UC Irvine Repository of Machine Learning Databases [Murphy & Aha] and originates from R. Detrano, Cleveland Clinic Foundation. Peter Dayan is thanked for comments on an earlier version of this paper.

References

Ahmad, S. & Tresp, V. (1993). Some solutions to the missing feature problem in vision. In *Advances in Neural Information Processing Systems*, volume 5, pp 393–400. Morgan Kaufmann, San Mateo, CA.

Baxt, W.G. (1990). Use of an artificial neural network for data analysis in clinical decision-making: The diagnosis of acute coronary occlusion. *Neural Computation*, **2(4)**, 480–489.

Chávez, Tom & Henrion, Max (1994). Efficient estimation of the value of information in Monte Carlo models. In *10th Conference on Uncertainty in Artificial Intelligence*, Seattle, WA.

Cohn, David A. (1994). Neural network exploration using optimal experiment design. In *Advances in Neural Information Processing Systems*, volume 6, pp 679–686. Morgan Kaufmann, San Mateo, CA.

Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series, B.*, **39**, 1–38.

Dobson, Annette J. (1990). *An Introduction to Generalized Linear Models.* Chapman and Hall, London, second edition.

Duda, R.O. & Hart, P.E. (1973). *Pattern Classification and Scene Analysis.* Wiley, New York.

Encyclopædia Britannica (1992). *The New Encyclopædia Britannica.* Encyclopaedia Britannica, Inc., Chicago, Ill., 15th edition. Britannica Online is at URL `http://www.eb.com/`.

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

Gennari, J.H., Langley, P. & Fisher, D. (1989). Models of incremental concept formation. *Artificial Intelligence*, **40**, 11–62.

Ghahramani, Z. & Jordan, M.I. (1994). Supervised learning from incomplete data via an EM approach. In *Advances in Neural Information Processing Systems*, volume 6, pp 120–127. Morgan Kaufmann, San Mateo, CA.

Gorry, G. Anthony & Barnett, G. Octo (1967). Experience with a model of sequential diagnosis. *Computers and Biomedical Research*, **1**, 490–507.

Heckerman, David E. (1991). *Probabilistic Similarity Networks.* MIT Press, Cambridge, MA.

Heckerman, D.E. & Nathwani, B.N. (1992). Toward normative expert systems: Part II. Probability-based representations for efficient knowledge acquisition and inference. *Methods of Information in Medicine*, **31**, 106–116.

Heckerman, D.E., Horvitz, E.J. & Nathwani, B.N. (1992). Toward normative expert systems: Part I. The Pathfinder project. *Methods of Information in Medicine*, **31**, 90–105.

Henrion, M., Breese, J.S. & Horvitz, E.J. (1991). Decision analysis and expert systems. *AI Magazine*, **12(4)**.

Hush, Don R. & Horne, Bill G. (1993). Progress in supervised neural networks — What's new since Lippmann? *IEEE Signal Processing Magazine*, pp 8–39, January.

Jacobs, R.A., Jordan, M.I., Nowlan, S.J. & Hinton, G.E. (1991). Adaptive mixtures of local experts. *Neural Computation*, **3(1)**, 79–87.

Kononenko, I. (1989). Bayesian neural networks. *Biological Cybernectics*, **61**, 361–370.

Ledley, Robert S. & Lusted, Lee B. (1959). Reasoning foundations of medical diagnosis. *Science*, **130(3366)**, 9–21.

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

McLachlan, Geoffrey J. & Basford, Kaye E. (1988). *Mixture Models: Inference and Applications to Clustering.* Marcel Dekker, Inc., New York, NY.

Miller, Randolph A., Pople, Harry E. & Myers, Jack D. (1982). Internist-1: An experimental computer-based diagnostic consultant for general internal medicine. *New England Journal of Medicine*, **307**, 468–476.

Murphy, P.M. & Aha, D.W. UCI repository of machine learning databases. Dept. of Information and Computer Science, University of California, Irvine, USA. URL `ftp://ics.uci.edu/pub/machine-learning-databases`.

Nowlan, Steven J. (1991). *Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures.* PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Pearl, Judea (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, **1**, 81–106.

Richard, M.D. & Lippmann, R.P. (1991). Neural network classifiers estimate Bayesian *a posteriori* probabilities. *Neural Computation*, **3(4)**, 461–483.

Shavlik, J.W., Mooney, R.J. & Towell, G.G. (1991). Symbolic and neural learning algorithms: An experimental comparison. *Machine Learning*, **6**, 111–143.

Shortliffe, E.H. (1976). *Computer-Based Medical Consultations: MYCIN.* Elsevier, New York, NY.

Stensmo, Magnus (1991). A query-reply classification system based on an artificial neural network. Technical Report TRITA-NA-9107, Dept. of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden.

Tou, Julius T. & Gonzales, Rafael C. (1974). *Pattern Recognition Principles.* Addison-Wesley, London. 4th printing 1981.

Tresp, V., Ahmad, S. & Neuneier, R. (1994). Training neural networks with deficient data. In *Advances in Neural Information Processing Systems*, volume 6, pp 128–135. Morgan Kaufmann, San Mateo, CA.

von Neuman, John & Morgenstern, Oscar (1947). *Theory of Games and Economic Behavior.* Princeton University Press, Princeton, NJ.

Xu, L. & Jordan, M.I. (1994). Theoretical and experimental studies of the EM algorithm for unsupervised learning based on finite Gaussian mixtures. Technical Report 9302, MIT Computational Cognitive Science.