Learning Hidden Markov Models with Geometric Information

Hagit Shatkay and Leslie Pack Kaelbling

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-97-04
April 1997

Learning Hidden Markov Models with Geometric Information

Hagit Shatkay Leslie Pack Kaelbling
Department of Computer Science
Brown University
Providence, RI 02912
{hs,lpk} @cs.brown.edu

Abstract

Hidden Markov models (HMMs) and partially observable Markov decision processes (POMDPs) provide a useful tool for modeling dynamical systems. They are particularly useful for representing environments such as road networks and office buildings, which are typical for robot navigation and planning. In a previous paper [SK97] we have empirically shown that by taking advantage of readily available odometric information, learning HMMs/POMDPs can be made faster and better. This paper extends our previous results in two ways. We show that our algorithm, which introduces an additional data structure and constraints to the standard HMM and extends the Baum-Welch algorithm to accommodate them, indeed converges to a local maximum of the likelihood function. In addition, we extend our empirical study, and show that the algorithm is robust in the face of reduction in the length of the training sequences. Thus, the use of local odometric information yields faster convergence to better solutions with less data.

1 Introduction

Hidden Markov models (HMMs), as well as their generalization to partially observable Markov decision processes (POMDPs), model a variety of nondeterministic dynamical systems as abstract probabilistic state-transition systems with discrete states, observations, and possibly actions. In this paper we concentrate on the special case of models in which states can be associated with points in a metric configuration space. These are appropriate in contexts such as office building, road network, or sewerage system modeling. Specifically, such POMDP models have proven particularly useful as a basis for robot navigation in buildings, providing a sound method for localization and planning [SK95, NPB95, CKK96]. Much of the previous work on planning assumed that the POMDP model is acquired manually; such manual acquisition can be very tedious and it is often difficult to obtain correct probabilities.

An ultimate goal is to be able to learn such models automatically, both for robustness and in order to cope with new and changing environments. Since POMDP models are a

simple extension of HMMs, they can, theoretically, be learned with a simple extension to the Baum-Welch algorithm [Rab89] for learning HMMs. However, without a strong prior constraint on the structure of the model, the Baum-Welch algorithm does not perform very well: it is slow to converge, requires a great deal of data, and often becomes stuck in local maxima.

Our work focuses on showing how weak metric information about the relationship between states can be used to significantly improve model learning. Such information is readily available in many cases and is often ignored during the process of learning topological maps. We have empirically shown [SK97] that the odometric ability of the robot, which allows measuring relative transformations between its configurations, can be used to learn maps of environments in a faster and more accurate fashion. In this paper we provide two extensions to the previous work. We *prove* that our algorithm, which introduces an additional matrix and geometrical constraints, is indeed a correct extension of the Baum-Welch algorithm in the sense that it converges to a local maximum of the likelihood function. In addition, we extend the empirical study to show that good results can be obtained with *less data*. That is, shorter data sequences that are insufficient for obtaining a good model using the standard Baum-Welch algorithm, suffice for learning significantly better models when the odometric information is taken into account. Thus, our algorithm yields faster convergence to better solutions with less data.

2 Related Work

This paper has two main related areas: map learning and correctness proofs for Baum-Welch extensions. A lot of work has been done on learning maps for mobile robotics and on learning stochastic models of dynamical systems in general. In this section, we focus on map learning for robots, and refer to closely related work in the first part of this section. A large number of papers is available, dealing with the the Baum-Welch algorithm and its extensions and proofs. We provide a brief survey of them in the second half of the section, and show their relevance in Section 4, where we present and prove our algorithm.

In the context of robot navigation, there is a distinction between two main types of maps: *metric* and *topological*. The former is the best choice when it is necessary for a robot to know its location accurately in terms of metric coordinates. However, in our environments of interest such as office buildings with corridors and rooms, or networks of roads, maps that simply specify the topology of important locations and their connections suffice. Such maps are typically less complex and support much more efficient planning than metric maps. Topological maps are flexible in allowing a more general notion of state, possibly including information such as the robot's battery voltage or whether or not it is holding a bagel. Some approaches learn an underlying deterministic map of the world, independent of the noise in the robot's sensing and action processes. We prefer to learn a combined model of the world and the robot's perception and interaction with the world; this allows robust planning that takes into account likelihood of error in sensing and action.

The work most closely related to ours is by Koenig and Simmons [KS96b, KS96a], who learn POMDP models (stochastic topological maps) of a robot hallway environment. They also recognize the impossibility of learning such a model without initial information; they

solve the problem by using a human-provided topological map, together with further constraints on the shared structure of the model. A modified version of the Baum-Welch algorithm learns the parameters of the model. They also developed an incremental version of Baum-Welch that allows it to be used on-line in certain kinds of environments. Their models contain very weak metric information, representing hallways as chains of 1-meter segments and allowing the learning algorithm to select the most probable chain length. This method is effective, but results in large models (size is proportional to hallways' length).

In contrast, we directly incorporate odometric information into the Baum-Welch algorithm. To prove the correctness of our extension to the Baum-Welch algorithm, in terms of local maximization of the likelihood, we follow the proof techniques given by Levinson et al [LRS83], Baum et al [BPS70] and Juang et al [JLS86].

Levinson et al provide an extensive and coherent introduction to learning standard HMMs with discrete states and observations. They follow Baum's proof for showing that the learned model locally maximizes the likelihood function $P(\text{data}|\text{model})$. They also show how to handle simple linear constraints specified over single parameters ($x_{ij} \geq \epsilon$). Baum et al [BPS70] provide and prove the reestimation formulae for HMMs with continuous observations, where the density function for the observations in each state must be log-concave. Their reestimation formulae were later proved by Dempster et al [DLR77] to be a special case of the Expectation Maximization family of algorithms. Baum's work was further extended by Liporace [Lip82], to the case of observations whose density function is multivariate and elliptically symmetric. Juang et al [Jua85, JLS86] provide an additional extension to handle multivariate general Gaussian mixtures, which are not necessarily symmetric.

In all of the above work the model consists of two matrices—the transition probability matrix and the observation distribution matrix, as well as an initial-state distribution vector. We introduce an additional matrix, the odometric relation matrix, into the model. The latter matrix has some geometrical constraints imposed on it, and we prove that our learning algorithm, which attempts to maintain these constraints, indeed converges to a local maximum likelihood model.

3 Models and Assumptions

In the following two sections, we describe the model and algorithms used for learning an HMM, rather than a POMDP. Extension to POMDPs is technically straightforward but notationally more cumbersome.

The world is composed of a finite set of states. The dynamics of the world are described by state-transition distributions that specify the probability of making transitions from one state to the next. There is a finite set of observations that can be made in each state; the frequency of such observations is described by a probability distribution and depends only on the current state. In our model, observations are *multi-dimensional*, so an observation is a vector of values, each chosen from a finite domain. It is assumed that observation values are conditionally independent, given the state. Each state is assumed to be associated with a (not necessarily unique) point in some metric space. Whenever a state

transition is made, the robot records an *odometry vector*, which estimates the location of the current state relative to the previous state. It is assumed that the components of the odometry vector are corrupted with independent normal noise (the extension to dependent noise is straightforward).

More formally, a model is a tuple $\lambda = \langle S, O, A, B, R, \pi \rangle$, where

- $S = \{s_1, \ldots, s_N\}$ is a finite set of N states;
- $O = \prod_{i=1}^{l} O_i$ is a finite set of observation vectors of length l; the ith element of an observation vector is chosen from the finite set O_i;
- A is a stochastic transition matrix, with $A_{i,j} = Pr(q_{t+1} = s_j | q_t = s_i)$; $1 \leq i, j \leq N$; q_t is the state at time t;
- B is an array of l stochastic observation matrices, with $B_{i,j,o} = Pr(V_t[i] = o | q_t = s_j)$; $1 \leq i \leq l,\ 1 \leq j \leq N,\ o \in O_j$; V_t is the observation vector at time t;
- R is a relation matrix, specifying for each pair of states, s_i and s_j, the mean and variance of the D-dimensional metric relation between them; $\mu_{ij}^k \stackrel{\text{def}}{=} \mu(R_{i,j}[k])$ is the mean of the kth component of the relation between s_i and s_j and $(\sigma_{ij}^k)^2 \stackrel{\text{def}}{=} \sigma^2(R_{i,j}[k])$, the variance, where $1 \leq k \leq D$. Furthermore, R is *geometrically consistent*: for each component k, the relation $R^k(a,b) \stackrel{\text{def}}{=} \mu(R_{a,b}[k])$ must be a *directed metric*, satisfying the following properties for all states a, b, and c:
  - $R^k(a,a) = 0$;
  - $R^k(a,b) = -R^k(b,a)$ *(anti-symmetry)*; and
  - $R^k(a,c) = R^k(a,b) + R^k(b,c)$ *(additivity)*;
- π is a stochastic initial probability vector describing the distribution of the initial state; for simplicity it is assumed here to be $\langle 1, 0, 0, \ldots, 0 \rangle$, implying that the robot is always started in state s_0.

A learning algorithm starts from an initial model λ_i and is given a sequence of *experience* E; it returns a revised model λ, with the goal of maximizing the likelihood $P(E|\lambda)$. P denotes, in this case, a probability density function. The experience sequence E is of length T; each element is a pair $\langle r_t, V_t \rangle$, where r_t is the observed relation vector between q_{t-1} and q_t and V_t is the observation vector at time t.

4 Algorithm

Our algorithm extends the Baum-Welch algorithm [Rab89] to deal with the relational information and the factored observation sets. The Baum-Welch algorithm has been proven to provide a monotonically increasing convergence of $P(E|\lambda)$ to a local maximum, for discrete observations as well as for continuous ones (with some restrictions), as discussed in Section 2. However, our extension introduces an additional component, namely, the relation (R) matrix. It can be interpreted as having two kinds of observations: *state*

observations (as in the ordinary HMM—with the distinction that our observations are integer vectors rather than integers) and *transition* observations (the odometry relations between states). The latter must satisfy geometrical constraints. Under such conditions an extension of the update formulae and a proof of their correctness is required. The proof is along the lines of those given by Baum et al [BPS70], and by Juang [Jua85], and is given in section 4.3.

4.1 Computing State-Occupation Probabilities

Following Rabiner [Rab89], we first compute the forward (α) and backward (β) matrices. When all measurements are discrete, $\alpha_t(i)$ is the probability of observing E_0 through E_t and $q_t = s_i$, given λ; and $\beta_t(i)$ is the probability of observing E_{t+1} through E_{T-1} given $q_t = s_i$ and λ. When some of the measurements are continuous (as is the case with R), these matrices contain probability density values rather than probabilities.

The forward procedure for calculating the α matrix is initialized with

$$\alpha_0(i) = \begin{cases} b_0^i & \text{if } \pi(i) = 1 \\ 0 & \text{otherwise} \end{cases} ,$$

and continued for $0 < t \leq T-1$ with

$$\alpha_t(j) = \sum_i \alpha_{t-1}(i) A_{i,j} f(r_t | R_{i,j}) b_t^j \ ,$$

where $f(r_t|R_{i,j})$ is the density of point r_t according to the normal distribution represented by the means and variances in entry i,j of the relation matrix R, and b_t^j is the probability of observing vector v_t in state s_j; that is, $b_t^j = \prod_{i=1}^{l} B_{i,j,v_t[i]}$.

The backward procedure for calculating the β matrix is initialized with

$$\beta_{T-1}(j) = 1 \ ,$$

and continued for $0 \leq t < T-1$ with

$$\beta_t(i) = \sum_j A_{i,j} f(r_{t+1} | R_{i,j}) b_{t+1}^j \beta_{t+1}(j) \ .$$

Given α and β, we now compute the state-occupation and state-transition probabilities (γ and ξ respectively). The state-occupation probabilities are computed as follows:

$$\begin{aligned} \gamma_t(i) &= \Pr(q_t = s_i | E, \lambda) = \frac{P(q_t = s_i, E|\lambda)}{P(E|\lambda)} \\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \ . \end{aligned} \tag{1}$$

Similarly, the state-transition probabilities are computed as:

$$\begin{aligned} \xi_t(i,j) &= \Pr(q_t = s_i, q_{t+1} = s_j | E, \lambda) \\ &= \frac{\alpha_t(i) A_{i,j} b^j{}_{t+1} f(r_{t+1}|R_{i,j}) \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) A_{i,j} b^j{}_{t+1} f(r_{t+1}|R_{i,j}) \beta_{t+1}(j)} \ . \end{aligned} \tag{2}$$

We note that the numerator and the denominator in the fractions are both density functions, but the quotient is a discrete probability function. These are essentially the same formulae appearing in [Rab89], but taking into account the density of the relational observation.

4.2 Updating Model Parameters

In this phase of the algorithm, the goal is to find a new model, λ, that maximizes $P(E|\lambda,\gamma)$. Generally, this is done by simple maximum-likelihood estimation of the probability distributions in A and B by computing expected transition and observation frequencies. It is more difficult in our model, because we must also compute a new relation matrix, R, under the constraint that it remain geometrically consistent. Through the rest of this section we use the notation $\overline{y}$ to denote a reestimated value, and y to denote the current value.

The A and B matrices can be straightforwardly re-estimated; $\overline{A}_{i,j}$ is the expected number of transitions from s_i to s_j divided by the expected number of transitions *from* s_i:

$$\overline{A}_{i,j} = \frac{\sum_{t=0}^{T-2} \xi_t(i,j)}{\sum_{t=0}^{T-2} \gamma_t(i)} \quad , \tag{3}$$

and $\overline{B}_{i,j,o}$ is the expected number of times observing o along the ith dimension when in s_j divided by the expected number of times in s_j:

$$\overline{B}_{i,j,o} = \frac{\sum_{t=0}^{T-1} I(V_t[i]=o)\gamma_t(j)}{\sum_{t=0}^{T-1} \gamma_t(i)} \quad , \tag{4}$$

where $I(c)$ is an indicator with value 1 if c is true and 0 otherwise.

If we were not going to enforce geometrical consistency, then the R matrix would be re-estimated by:

$$\overline{\mu}_{ij}^k = \mu(\overline{R}_{i,j}[k]) = \frac{\sum_{t=0}^{T-2} r_t[k]\xi_t(i,j)}{\sum_{t=0}^{T-2} \xi_t(i,j)} \tag{5}$$

$$(\overline{\sigma}_{ij}^k)^2 = \sigma^2(\overline{R}_{i,j}[k]) = \frac{\sum_{t=0}^{T-2} (r_t[k]-\overline{\mu}_{ij}^k)^2\xi_t(i,j)}{\sum_{t=0}^{T-2} \xi_t(i,j)} \quad . \tag{6}$$

In the current implementation, we enforce only two of the three constraints of geometrical consistency. Zero distances between states and themselves are trivially enforced. Antisymmetry is enforced by using the data from s_j to s_i as well as from s_i to s_j when reestimating $\mu(R_{i,j})$. Thus the actual reestimation formula for $\mu(R_{i,j})$ is:

$$\overline{\mu}_{i,j}^k = \frac{\sum_{t=0}^{T-2} [r_t[k]\xi_t(i,j) - r_t[k]\xi_t(j,i)]}{\sum_{t=0}^{T-2} [\xi_t(i,j) + \xi_t(j,i)]} \quad . \tag{7}$$

The additivity constraint is not currently enforced through the updates, though it is satisfied in the initial models, thus biasing the learned model towards satisfying it. We are currently exploring some relaxation techniques, based on spring systems, for enforcing the additivity constraint. The complexity of the algorithm per iteration is still $O(TN^2)$, like the standard Baum-Welch algorithm.

4.3 Proof of the Reestimation Formulae

For this type of reestimation algorithm, proving the correctness of the reestimation formulae means proving that through repeated reestimation, the algorithm converges to a fixed point model $\overline{\lambda}$, such that $\overline{\lambda}$ is a local maximum of the likelihood function $P(E|\lambda)$, where E is the observed experience sequence.

There are several proof techniques for the correctness of the reestimation formulae for the standard Baum-Welch algorithm (under various kinds of observation matrix B) [BPS70, DLR77, LRS83, Jua85, JLS86]. Our proof uses the same approach as the latter two. It is straightforward to show that maximization of the likelihood function with respect to each of the parameters separately is equivalent to its maximization with respect the complete model. Hence, we break the proof, and prove that each of the reestimation formulae indeed maximizes the likelihood function with respect to the associated reestimated parameter.

4.3.1 Transitions and Observations

To prove the correctness of formulae (3) and (4), we use the central theorem of Baum and Sell [BS68], which claims[1] that *for $x = \{x_{ij}\}$ s.t. $x_{ij} > 0$, $1 \le j \le N$, and $\sum_{j=1}^{N} x_{ij} = 1$, given a homogeneous polynomial P in the variable x_{ij}, with nonnegative coefficients, the transformation*

$$\overline{x_{ij}} = \frac{x_{ij}\frac{\partial P}{\partial x_{ij}}}{\sum_{k=1}^{N} x_{ik}\frac{\partial P}{\partial x_{ik}}} \tag{8}$$

satisfies $P(x) \le P(\overline{x})$, and $\overline{x} = x$ if and only if x is a local maximum of P.

The density expression $P(E|\lambda) = \sum_{i=1}^{N} \alpha_{T-1}(i) = \sum_{i=1}^{N} \alpha_{T-1}(i)\beta_{T-1}(i)$, which we want to maximize, is indeed a homogeneous polynomial in A_{ij} and in B_{ijo}. Both A_{ij} and B_{ijo} are discrete probability distributions, therefore are positive and satisfy $\sum_{j=1}^{N} A_{ij} = 1$ and $\sum_{o \in O_i} B_{ijo} = 1$. Hence, according to the above theorem the reestimation formula for A_{ij}, that leads to a local maximization of $P(E|A_{ij})$ is:

$$\overline{A_{ij}} = \frac{A_{ij}\frac{\partial P(E|\lambda)}{\partial A_{ij}}}{\sum_{k=1}^{N} A_{ik}\frac{\partial P(E|\lambda)}{\partial A_{ik}}} \ . \tag{9}$$

We now need to show that the right-hand side of formula (9) is equal to that of (3). To do this we need to show that:

$$\frac{\partial P(E|\lambda)}{\partial A_{ij}} = \sum_{t=0}^{T-2} \alpha_t(i) b^j{}_{t+1} f(r_{t+1}|R_{i,j})\beta_{t+1}(j) \ . \tag{10}$$

By substituting the right hand side expression into (9), and using equations (1,2) we get the desired equality.

[1] The theorems in the paper are actually somewhat stronger and the statement below is just one consequence.

By induction on m, $0 \leq m \leq T-1$, it is easy to show that:

$$\sum_{i=1}^{N} \frac{\partial \alpha_m(i)}{\partial A_{ij}} \beta_m(i) = \sum_{t=0}^{m-1} \alpha_t(i) b^j{}_{t+1} f(r_{t+1}|R_{i,j}) \beta_{t+1}(j) \ .$$

For $m = T-1$ we get (10), which concludes the proof of formula (3). The proof for B_{ijo} (formula (4)) is almost identical.

4.3.2 Odometric Relations

We note that the density expression, $P(E|\lambda)$, is not a polynomial in $\mu(R_{i,j}[k])$ and $\sigma^2(R_{i,j}[k])$. Hence the above theorem can not be applied here. Instead, we use the technique of maximizing Baum's auxiliary function, following Section 4 of the paper by Baum et al [BPS70]. We shall denote by λ_x, where x is some relation matrix, the model whose A and B matrices are the same as those of λ, but whose relation matrix R is replaced by the matrix x.

We start by making the observation that if $\mathcal{S}$ is the set of all state sequences of length T, i.e. $\mathcal{S} = \{S\}$ where $S = s_0, \ldots, s_{T-1}$ is a sequence of states of length T, the density $P(E|\lambda)$ can be expressed as

$$P(E|\lambda) = \sum_{S \in \mathcal{S}} \tau(S) \prod_{t=1}^{T-1} f(r_t|R_{s_{t-1},s_t}),$$

where $\tau(S)$ is a product of initial, transition and observation probabilities. Recalling that $f(r_t|R_{i,j})$ denotes the density of r_t according to the D-variate independent[2] normal distribution with the parameters stored in $R_{i,j}$, we rewrite it as

$$f(r_t|R_{i,j}) = \prod_{k=1}^{D} \frac{f_{ij}^k(r_t^k)}{\sqrt{2\pi}\sigma_{ij}^k}, \tag{11}$$

where $r_t^k = r_t[k]$ and $f_{ij}^k(r_t^k) = e^{-(r_t^k - \mu_{ij}^k)^2/2(\sigma_{ij}^k)^2}$. We also use the notation: $f_{ij}(r_t) = \prod_{k=1}^{D} f_{ij}^k(r_t^k)$.

Baum et al [BPS70] introduce an auxiliary function, Q, and prove that maximizing it is the same as increasing the likelihood. More formally, the $\overline{R}$ that maximizes the auxiliary function

$$\begin{aligned} Q(R, \overline{R}) &= \sum_{S \in \mathcal{S}} P(E, S|\lambda) \log\Big(\prod_{t=1}^{T-1} f(r_t|\overline{R}_{s_{t-1},s_t})\Big) \\ &= \sum_{S \in \mathcal{S}} P(E, S|\lambda) \sum_{t=1}^{T-1} \log(f(r_t|\overline{R}_{s_{t-1},s_t})) \end{aligned}$$

also satisfies $P(E|\lambda_{\overline{R}}) \geq P(E|\lambda_R)$.

Since the $\overline{R}$ that maximizes $Q(R, \overline{R})$ also maximizes the same expression in which $\sqrt{2\pi} f_{ij}^k(r_t^k)$ is substituted for $f_{ij}^k(r_t^k)$, we can ignore the $\sqrt{2\pi}$ factor in (11) and rewrite Q as

$$Q(R, \overline{R}) = \sum_{S \in \mathcal{S}} P(E, S|\lambda) \sum_{t=1}^{T-1} \sum_{k=1}^{D} [\log(\overline{f}^k_{s_{t-1},s_t}(r_t^k)) - \log(\overline{\sigma}^k_{s_{t-1},s_t})] \ . \tag{12}$$

[2]The independence assumption is not necessary, but it makes for a clearer proof.

Since the normal distribution is strictly log-concave, a slight adaptation to the proof of Theorem 4.1 in [BPS70] is sufficient for showing that Q above has a unique global maximum as a function of $\overline{\mu}_{ij}^k$ and $\overline{\sigma}_{ij}^k$, which is the unique point in which the partial derivatives of Q according to $\overline{\mu}_{ij}^k$ and $\overline{\sigma}_{ij}^k$ are 0. In the rest of this section we show that the reestimation formulae, (5) and (6) indeed find the maximizing $\overline{\mu}_{ij}^k$ and $\overline{\sigma}_{ij}^k$.

For a pair of states i, j and an odometry component k we can express the restriction of the auxiliary function Q to transitions from i to j and to the k^{th} observation component k as

$$Q_{ij}^k(R, \overline{R}) = \sum_{S \in \mathcal{S}} P(E, S|\lambda) \sum_{\substack{t \text{ s.t.} \\ s_{t-1}=i \\ s_t=j}} (\log(\overline{f}^k_{s_{t-1},s_t}(r_t^k)) - \log(\overline{\sigma}^k_{s_{t-1},s_t})) \ .$$

Observing that $\xi_{t-1}(i,j) = \sum_{\substack{S \in \mathcal{S} \text{ s.t.} \\ s_{t-1}=i \\ s_t=j}} P(E, S|\lambda)$ allows us to rewrite $Q_{ij}^k(R, \overline{R})$ as

$$Q_{ij}^k(R, \overline{R}) = \sum_{t=0}^{T-2} \xi_t(i,j)(\log(\overline{f}^k_{i,j}(r_{t+1}^k)) - \log(\overline{\sigma}_{ij}^k)) \ . \tag{13}$$

Since $\frac{\partial Q}{\partial \overline{\mu}_{ij}^k} = \frac{\partial Q_{ij}^k}{\partial \overline{\mu}_{ij}^k}$ and $\frac{\partial Q}{\partial \overline{\sigma}_{ij}^k} = \frac{\partial Q_{ij}^k}{\partial \overline{\sigma}_{ij}^k}$, showing that the partial derivatives of Q_{ij}^k with respect to $\overline{\mu}_{ij}^k$ and $\overline{\sigma}_{ij}^k$ are 0 whenever equations (5) and (6) are satisfied, concludes our proof. The differentiation is straightforward and therefore is not given here.

4.3.3 Reestimation Proof for the Constrained Odometric Relations

We prove the correctness of formula (7) through the use of Lagrange multipliers for the maximization of the auxiliary function Q.

From equations (12) and (13) we have:

$$Q(R, \overline{R}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{D} Q_{ij}^k(R, \overline{R}) \ .$$

Any $\overline{\mu}_{ij}^k$ that maximizes Q also maximizes $2Q$. Thus, maximizing Q is the same as maximizing

$$\begin{aligned} 2Q(R, \overline{R}) &= 2 \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{D} Q_{ij}^k(R, \overline{R}) \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{D} Q_{ij}^k(R, \overline{R}) + \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{D} Q_{ji}^k(R, \overline{R}) \\ &= \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{D} (Q_{ij}^k(R, \overline{R}) + Q_{ji}^k(R, \overline{R})) \ . \end{aligned} \tag{14}$$

For maximizing the sum in (14) with respect to $\overline{\mu}_{ij}^k$ it is sufficient to maximize the sum of each pair $(Q_{ij}^k(R, \overline{R}) + Q_{ji}^k(R, \overline{R}))$ with respect to $\overline{\mu}_{ij}^k$, since all other pairs do not

contain expressions with $\overline{\mu}_{ij}^k$. Moreover, we can omit the factor $2(\overline{\sigma}_{ij}^k)^2$ from the exponent expression in

$$\overline{f}_{ij}^k(r_t^k) = e^{-(r_t^k - \overline{\mu}_{ij}^k)^2)/2(\overline{\sigma}_{ij}^k)^2} \quad .$$

Thus we are left with the following expression for maximization:

$$L_{ij}^k = \sum_{t=0}^{T-2} (\xi_t(i,j)(\log(e^{-(r_t^k - \overline{\mu}_{ij}^k)^2)} - \log(\overline{\sigma}_{ij}^k)) + \xi_t(j,i)(\log(e^{-(r_t^k - \overline{\mu}_{ji}^k)^2)} - \log(\overline{\sigma}_{ji}^k))) \quad , \quad (15)$$

under the constraint that $\overline{\mu}_{ij}^k + \overline{\mu}_{ji}^k = 0$.

Using the method of Lagrange multipliers, there exists a scalar λ_1 such that at each extremum point the following equations hold:

$$\begin{aligned}
\frac{\partial L_{ij}^k}{\partial \overline{\mu}_{ij}^k} &= \lambda_1 \frac{\partial(\overline{\mu}_{ij}^k + \overline{\mu}_{ji}^k)}{\partial \overline{\mu}_{ij}^k} = \lambda_1 \cdot 1 = \lambda_1 \\
\frac{\partial L_{ij}^k}{\partial \overline{\mu}_{ji}^k} &= \lambda_1 \frac{\partial(\overline{\mu}_{ij}^k + \overline{\mu}_{ji}^k)}{\partial \overline{\mu}_{ji}^k} = \lambda_1 \cdot 1 = \lambda_1 \\
\overline{\mu}_{ij}^k + \overline{\mu}_{ji}^k &= 0 \quad .
\end{aligned}$$

Solving this system of equations results in the formula given in (7), which concludes our proof. We note that for the special case where $i = j$ the value $\overline{\mu}_{ij}^k$ is 0, which shows that the update formula indeed satisfies two of the three geometrical consistency constraints.

4.4 Finding an Initial Model

It is typical in instances of the Baum-Welch algorithm to simply initialize the model at random, perhaps trying multiple initial models to find different local likelihood maxima. We have tried random initial models, as well as starting from a more informed model, based on the odometry information. In both the random and informed initializations the initial R matrix satisfies all three properties of geometrical consistency.

To build an informed model, we begin by assigning global metric coordinates to each element in the sequence E. This is done by accumulating the observed relations between consecutive pairs of states. This data set (ignoring other observation information) is fed into a simple k-means clustering algorithm, yielding a clustering of the data into N clusters. The clusters are taken to be the states, and the observations associated with a given cluster are interpreted as having been generated in the associated state. We then compute γ and ξ values, with the γ values being 0 or 1, since we use a deterministic clustering algorithm (it might be beneficial to use a stochastic clustering algorithm, such as Autoclass [CKS90]). The A, B, and R matrices are all estimated from γ and ξ as described in the previous section. Finally, an *ad hoc* process is used to adjust R to satisfy the additivity constraint.

5 Experiments

The motivation for this work is to take advantage of readily available odometry information, for learning topological models that are good, in less time while using less data. In a previous paper [SK97] we have demonstrated that the use of odometry readings significantly reduces the number of iterations required for convergence, and results in significantly better models both in terms of the resulting geometric map and in terms of the measured KL-divergence from the true model.

We extend the results here to show that while the use of shorter data sequences significantly effects the behavior of the standard Baum-Welch algorithm, our algorithm is pretty robust to reductions in the amount of data.

In the rest of this section we describe the experimental setting in a simulated robot domain. For the sake of completeness we briefly survey the results from our previous paper [SK97], and then empirically show that our algorithm performs well with less data.

5.1 Robot Domain

The robot moves through the corridors in an office environment. Low-level software provides a level of abstraction that allows the robot to move through hallways from intersection to intersection and to turn ninety degrees to the left or right. At each intersection, ultrasonic data interpretation allows the robot to perceive, in each of the four cardinal directions, whether there is an open space, a door, a wall, or something unknown. Of course, both the action and perception routines are subject to error. Finally, the robot has encoders on its wheels that allow it to estimate the robot's pose (position and orientation) with respect to its pose at the previous intersection.

For simplicity, we learn an HMM rather than a POMDP model. In our simulation, we manually generated an HMM representing a prescribed path of the robot through the complete office environment, consisting of 44 states, and the associated transition, observation, and odometric distributions. Figure 1 shows the true HMM corresponding to the hallway environment[3]. Black dots represent the physical locations of states. In most cases, multiple states (depicted as numbers in the plot) are associated with each location, corresponding to several possible orientations of the robot at that location. Solid arrows represent the most likely transitions (corridors traversed in a certain direction) between the locations. Dashed arrows represent less likely transitions between states, whose probability is 0.2 or higher. The larger black circle represents the initial state. Note that the length of the arrows is significant and represents the length of the corridors, drawn to scale.

5.2 Evaluation Method

Traditionally, in such experiments, the learned model is *quantitatively* compared to the actual model that generated the data. Each of the models induces a probability distribution on strings of observations; the asymmetric Kullback-Leibler divergence [KL51] between the two distributions is a measure of how good the learned model is with respect

[3]Observations are omitted for clarity.

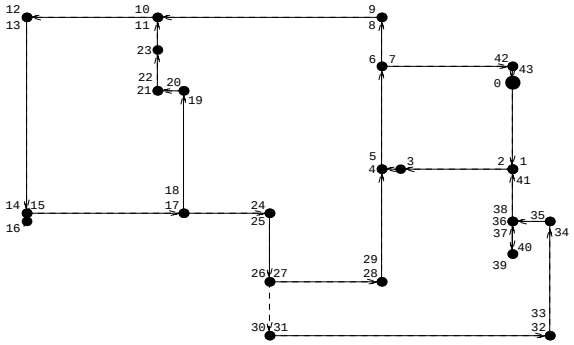

Figure 1: True map of hallway environment.

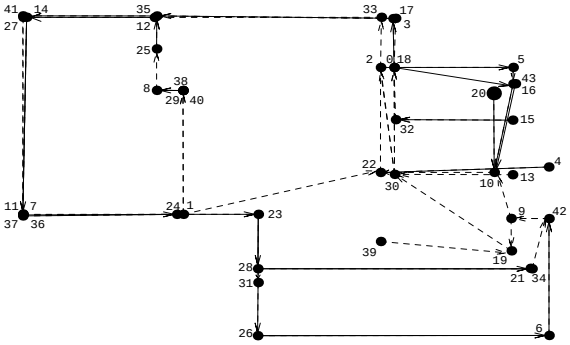

Figure 2: Nominal learned map of hallway environment.

to the true model. The smaller this measure is, the closer the learned model is to the true one, and the better is the result. We report our results in terms of a sampled version of the KL-divergence, as described by Rabiner [Rab89]. It is based on generating at random sequences of sufficient length (5 sequences of 1000 observations in our case) according to the distribution induced by the true model, and comparing their likelihoods according to the learned model with the true model likelihoods. We ignore the odometry information when applying the KL measure, thus allowing comparison between models that are learned with and without odometry.

Another way to evaluate the learned model is based on the *qualitative* plausibility of that model. In our domain, there are transitions and observations that usually take place, and are therefore more likely than the others. We can look at the most likely transitions and observations from each state and ask whether they correspond to the true underlying map of the world. Furthermore, the relational information gives us a rough estimate of the metric locations of the states with respect to each other. Figure 2 is such a nominal version of a learned map.

5.3 Results

We started by generating 5 data sequences, each of length 1000, using Monte Carlo sampling from the model depicted in Figure 1. We used three different settings of the learning algorithm:

- starting from an informed (cluster based) initial model and using odometry information;
- starting from a random initial model and using odometry information;
- starting from a random initial model without using odometry information (standard Baum-Welch).

Seq. #	Informed		Random		No Odo	
	KL	Iter. #	KL	Iter. #	KL	Iter. #
1	2.347	93.40	3.797	110.00	10.129	201.80
2	2.338	84.00	2.630	82.00	11.400	213.80
3	2.572	85.80	3.086	150.00	11.583	195.40
4	3.185	85.20	3.532	114.20	11.810	224.00
5	4.154	116.20	3.877	102.40	12.258	156.80

Table 1: Average results of three learning settings with five training sequences.

For showing faster convergence to better models, we ran the algorithm, for each sequence and each of the three algorithmic settings, repeatedly 5 times. (Note that there is non-determinism even when using informed initial models, since the clustering starts with random k-means, thus multiple runs give multiple results). In all the experiments, N was set to be 44, which is the "correct" number of states.

Figure 2 shows the nominal version of one learned map for a representative run. All arrows represent transitions that have probability 0.2 or higher. Solid arrows represent the most likely transition from each state and dashed arrows the less likely ones[4]. The length of the arrows represents the learned odometric relations. There is an obvious correspondence between groups of states in the learned and true models, and most of the transitions were learned correctly.

Table 1 lists the KL-divergence between the true and learned model, as well as the number of runs until convergence was reached, for each of the 5 sequences under each of the 3 learning settings, averaged over 5 runs per sequence. From the table it is clear that the KL-divergence with respect to the true model for models learned using odometry, starting from either an informed or a random initial model, is about *4 times smaller* than for models learned without odometry data. In addition, the number of iterations required for convergence when learning using odometry information is roughly half that required when ignoring odometry information. We used the t-test to verify the statistical significance of these results. Deeper analysis of both the qualitative and the quantitative results is given in [SK97].

To examine the influence of the amount of data on the quality of the learned models, we took one of the 5 sequences (Seq. #1) and used its prefixes of length 100 to 1000 (the complete sequence), in increments of 100, as individual sequences. We ran each of the three algorithmic settings over each of the 10 prefix sequences, 10 times repeatedly. We then used the KL-divergence as described above to evaluate each of the resulting models with respect to the true model. For each prefix length we averaged the KL-divergence over the 10 runs.

Table 2 summarizes the results of this experiment. It lists the mean KL-divergence over the 10 runs for each of the prefixes, as well as the standard deviation around this mean. Entries with KL-divergence $\rightarrow \infty$ indicate that sequences generated by the true model were assigned negligible (~0) probability by the learned model, which corresponds to an infinite KL-divergence. The plot in Figure 3 depicts the KL-divergence as a function of the sequence length for each of the three settings. Both the table and the plot demonstrate

[4]Bold dashed arrows represent transitions that are *almost* as likely as the most likely.

Seq. length	Informed		Random		No Odo	
	Mean KL	Std. Dev.	Mean KL	Std. Dev.	Mean KL	Std. Dev.
1000	2.097	0.71	3.016	1.24	10.139	1.90
800	1.805	0.28	2.954	1.75	13.319	1.26
600	2.353	1.14	2.569	1.36	14.542	1.62
400	2.953	0.60	2.801	1.69	22.714	4.69
300	3.998	1.53	3.892	3.50	$\to \infty$	NA
200	9.169	3.09	7.489	5.25	$\to \infty$	NA
100	$\to \infty$	NA	$\to \infty$	NA	$\to \infty$	NA

Table 2: Average results of three learning settings with 10 incrementally longer sequences .

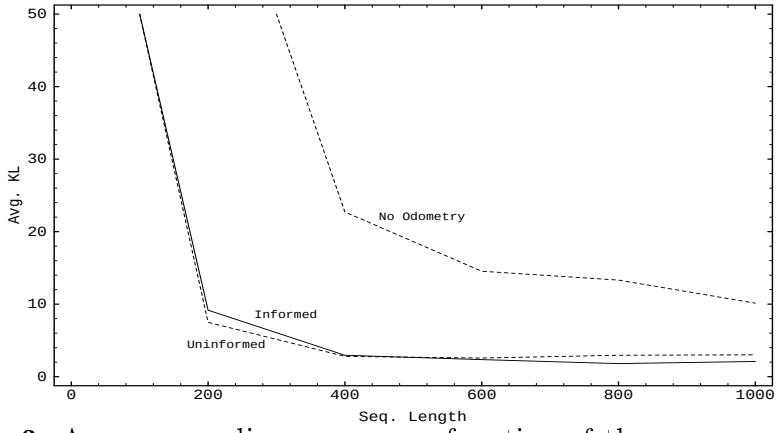

Figure 3: Average KL-divergence as a function of the sequence length

that, in terms of the KL-divergence, learning without the use of odometry is much more sensitive to reduction in the amount of data than learning while using the odometry information.

We used the two-sample t-test to verify the statistical significance of these results. For example, the KL-divergence being greater for sequences of length 800 than for sequences of length 1000 when learning without the use of odometry is highly statistically significant ($p \ll 0.005$). In contrast, the KL-divergence is not statistically significantly greater when the odometry is used, for either informed or uninformed models. The informed model is even somewhat better, (with moderate statistical significance, $p < 0.11$), when using the sequence of length 800, due to better clustering when there is less accumulated noise on the odometry data.

We note that the data sequence is twice as "wide" when odometry is used than when it is not, that is, there are twice as many information "bits" in each point of the sequence when odometry information is recorded. (This is because in our case both the odometry readings vectors and the observation vectors are 3-dimensional). However, if this was the only justification for the gap in the KL-divergence when learning with and without odometry, the KL-divergence for models learned *with* odometry from sequences of length *400* should have been about the same as that for models learned *without* odometry from sequences of length *800*. Obviously, this is not the case. This strengthens our claim that the utilization of odometry significantly improves the learning itself.

6 Conclusions

Odometric information, which is often readily available, makes it possible to learn hidden Markov models efficiently and effectively, while using less data. The odometric information can be incorporated into the traditional HMM model, in a straightforward way, that guarantees convergence of the reestimation algorithm to a local maximum of the likelihood function.

If we are interested in learning the geometric relationships between states, using the odometry readings is obviously very helpful. Moreover, our experiments show that even when we are only interested in the underlying topological model, using odometry can both reduce the number of iterations required by the algorithm and improve the resulting model, while requiring shorter data sequences.

The work described in this paper is still in its preliminary stages. In the very near future, we will extend the example to learn the fully controllable POMDP rather than the HMM, and will train the algorithm with real robot data. It will be useful to investigate more sophisticated clustering methods for an initial informed model, to replace the naive clustering method used now. Finally, we would like to find an improved reestimation step that allows us to satisfy the additivity constraint, while still guaranteeing the convergence of this algorithm.

References

[BPS70] L. E. Baum *et al.*, *A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains*, The Annals of Mathematical Statistics, 41 , no. 1, 164–171, 1970.

[BS68] L. E. Baum and G. R. Sell, *Growth Transformations for Functions on Manifolds*, Pacific Journal of Mathematics, 27 , no. 2, 211–227, 1968.

[CKK96] A. R. Cassandra, L. P. Kaelbling and J. A. Kurien, *Acting Under Uncertainty: Discrete Bayesian Models for Mobile-Robot Navigation*, In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 1996.

[CKS90] P. Cheeseman *et al.*, *AutoClass: A Bayesian Classification System*, In J. W. Shavlik and T. G. Dietterich, editors, Readings in Machine Learning, pp. 296–306. Morgan-Kaufmann, 1990.

[DLR77] A. P. Dempster, N. M. Laird and D. B. Rubin, *Maximum Likelihood from Incomplete Data via the EM Algorithm*, Journal of the Royal Statistical Society, 39 , no. 1, 1–38, 1977.

[JLS86] B. H. Juang, S. E. Levinson and M. M. Sondhi, *Maximum Likelihood Estimation for Multivariate Mixture Observations of Markov Chains*, IEEE Transactions on Information Theory, 32 , no. 2, March 1986.

[Jua85] B. H. Juang, *Maximum Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains*, AT&T Technical Journal, 64 , no. 6, July-August 1985.

[KL51] S. Kullback and R. A. Leibler, *On Information and Sufficiency*, Annals of Mathematical Statistics, 22 , no. 1, 79–86, 1951.

[KS96a] S. Koenig and R. G. Simmons, *Passive Distance Learning for Robot Navigation*, In Proceedings of the Thirteenth International Conference on Machine Learning, pp. 266–274, 1996.

[KS96b] S. Koenig and R. G. Simmons, *Unsupervised Learning of Probabilistic Models for Robot Navigation*, In Proceedings of the IEEE International Conference on Robotics and Automation, 1996.

[Lip82] L. A. Liporace, *Maximum Likelihood Estimation for Multivariate Observations of Markov Sources*, IEEE Transactions on Information Theory, 28 , no. 5, 1982.

[LRS83] S. E. Levinson, L. R. Rabiner and M. M. Sondhi, *An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition*, The Bell System Technical Journal, 62 , no. 4, 1035–1074, 1983.

[NPB95] I. Nourbakhsh, R. Powers and S. Birchfield, *DERVISH: An Office-Navigating Robot*, AI Magazine, 16 , no. 1, 53–60, 1995.

[Rab89] L. R. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of the IEEE, 77 , no. 2, 257–285, February 1989.

[SK95] R. G. Simmons and S. Koenig, *Probabilistic Navigation in Partially Observable Environments*, In Proceedings of the International Joint Conference on Artificial Intelligence, 1995.

[SK97] H. Shatkay and L. P. Kaelbling, *Learning Topological Maps with Weak Local Odometric Information*, Submitted to IJCAI97, 1997.