Heading in the Right Direction

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

Abstract

Stochastic topological models, and hidden Markov models in particular, are a useful tool for robotic navigation and planning. In previous work we have shown how weak odometric data can be used to improve learning topological models, overcoming the common problems of the standard Baum-Welch algorithm. Odometric data typically contain directional information, which imposes two difficulties: First, the cyclicity of the data requires the use of special circular distributions. Second, small errors in the heading of the robot result in large displacements in the odometric readings it maintains. The cumulative rotational error leads to unreliable odometric readings. In the paper we present solutions to these problems by using a circular distribution and relative coordinate systems. We validate their effectiveness through experimental results from a model-learning application.

1 INTRODUCTION

Directional data is information consisting of magnitude and *direction*. Such data is an integral part of important applications in various areas of computer science in general and artificial intelligence in particular. In computer graphics, automatic production of pen-and-ink drawings and the production of animation based on magnetic trackers data requires statistical manipulation of directional data. In cognitive science, modeling routes chosen by animals [4] requires a similar kind of statistical manipulation. In the area of machine learning we often use probabilistic models for robot movement. Most aspects of robot movement (arm movement as well as the whole body movement) can be described in terms of location and heading change, requiring the use and manipulation of directional data.

Probabilistic models are widely used within the AI community. Such models may allow continuous probabilities, as demonstrated in work on Bayesian networks [7], hidden Markov models [5, 8], probabilistic clusters [2] and stochastic maps [19], to name a few. However, the assumption underlying all the above work is that continuous distributions are *linear* — that is — distributions that assign density to each point on the real line so that the area under the density curve, integrated over the whole real line, is 1.[1] Such models do not take into account directional data, which is inherently *cyclic*. Under circular distributions the density of any point x on the real line is the same as that of $x + k\xi$ where k is any integer and ξ is some real number.

The need for circular distributions has long been realized by statisticians [6], but the practice of using them has not found its way into the computer science community and to the machine learning community in particular. One of the goals of this paper is to point out the usefulness of one specific circular distribution in the context of robotics, and provide a short tutorial on circular distributions.

Another special aspect of directional data is its sensitivity to errors. As most navigators, pilots and skippers have experienced, a small angular deviation from the original course causes a big displacement at the final location. This problem is very prominent in mobile robots, where drifts and drags of the wheels and disalignment of both engines and floors can cause a robot to face in the wrong heading with respect to its own odometric readings. Odometric information is recorded by the robot along three dimensions; it consists of the changes along the x and the y axis as well as a change in the *heading* of the robot within a global coordinate system. In our previous work on learning topological models [17] we made several assumptions about the odometric data:

- All odometric measures are normally distributed.

[1] Most often the distribution is Gaussian.

- All corridors are perpendicular to each other.
- The robot, when collecting the data, is using the perpendicularity assumption, and is collecting the data with respect to one global coordinate system.

This paper demonstrates the problematic aspects of these assumptions and introduces our solution to the problems, together with preliminary results that demonstrate the effectiveness of our solution. The rest of the paper is organized as follows: Section 2 describes our application and motivates the need for circular distributions in the context of machine learning; Section 3 presents the von Mises distribution, which is a circular version of the normal distribution; Section 4 discusses the problems faced due to heading deviations and presents our solution to the problem; Section 5 presents experiments and results to demonstrate the usefulness of our approach; Section 6 concludes the paper.

2 LEARNING TOPOLOGICAL MODELS

Hidden Markov models (HMMs), as well as their generalization to models for partially observable Markov decision processes (POMDP models), are a useful tool for representing environments such as road networks and office buildings, which are typical for robot navigation and planning [1, 14, 18]. Previous work on planning with such models typically assumed that the model is manually provided. Manual acquisition of these models can be very tedious and hard. It is desirable 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 [15] 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. In previous work [16, 17] we demonstrated how the simple Baum-Welch algorithm can be enhanced with weak local odometric information to learn better models faster, under the assumption listed above. For the sake of completeness, we briefly review the essentials of this work here.

A 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 turn ninety degrees to the left or right. At each intersection, ultrasonic data interpretation lets the robot observe, in each of the four cardinal directions, whether there is an open space, a door, a wall, or something unknown. The robot also has encoders on its wheels that allow it to estimate its current pose (position and orientation) with respect to its pose at the previous intersection. Of course, the action and perception routines and the odometric measures are all subject to error. The learning task is to deduce a model from the recorded observations and odometric information.

Our learning algorithm gets as an input an experience sequence E of observations and odometric readings, and produces as output an HMM[2], λ, of the environment, such that the likelihood, $Pr(\mathsf{E}|\lambda)$, is locally maximized. Formally, the standard HMM is defined as a tuple $\lambda = \langle S, O, A, B, \pi \rangle$, 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 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;
- π is a stochastic initial probability vector describing the distribution of the initial state.

Odometric information gathered by the robot is not an inherent part of the topological model, but is used by the learning algorithm to better identify and distinguish states. To facilitate the use of this information we augment the standard model with the odometric relation matrix:

- 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; $\mu_{ij}^d \stackrel{\text{def}}{=} \mu(R_{i,j}[d])$ is the mean of the d^{th} component of the relation between s_i and s_j and $(\sigma_{ij}^d)^2 \stackrel{\text{def}}{=} \sigma^2(R_{i,j}[d])$, the variance, where $1 \leq d \leq D$. Furthermore, R is *geometrically consistent*: for each component d, the relation $R^d(a,b) \stackrel{\text{def}}{=} \mu(R_{a,b}[d])$ must satisfy the following properties for all states a, b, and c:
  - ⋄ $R^d(a,a) = 0$;
  - ⋄ $R^d(a,b) = -R^d(b,a)$ *(anti-symmetry)*; and
  - ⋄ $R^d(a,c) = R^d(a,b) + R^d(b,c)$ $(additivity)$;

The odometric information recorded by the robot at time t, r_t, consists of the change in the x and y coordinates of the odometric readings when moving from state q_{t-1} to state q_t, as well as the change of the robot's heading, θ, between these states.

An arbitrary initial model λ_0 is assumed. Then an expectation maximization algorithm [3] is executed as follows:

[2]We discuss here HMMs rather than POMDP models. Extension to POMDPs is straightforward, but notationally more cumbersome.

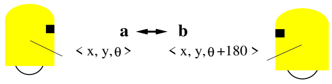

Figure 1: Robot changes heading from state *a* to state *b*.

- *E-step*: computes the state-occupation and transition probabilities, $\gamma_t(i) = Pr(q_t = s_i|\mathsf{E}, \lambda)$ and $\xi_t(i,j) = Pr(q_t = s_i, q_{t+1} = s_j|\mathsf{E}, \lambda)$, respectively, at each time t in the sequence, given E and the current model λ, and
- *M-step*: finds a new model λ that maximizes $\Pr(\mathsf{E}|\lambda, \gamma, \xi)$.

Introducing odometric information requires iterative updates of the odometric relations between pairs of states, in the relation matrix, R. The updates need to maintain the properties listed above, although currently the update procedure only satisfies the first two.

The learning task is further complicated by the special nature of the heading reading and the rotational errors accrued. The following section goes in more detail into the special issues of handling the heading information. The rest of the paper deals with resolving the problems caused by rotational errors.

3 DIRECTIONAL DATA AND DISTRIBUTIONS

Suppose a robot is in state a, which is in location $\langle x, y\rangle$ facing in direction θ, as shown in figure 1. By turning backwards, it transitions to state b, and a respective change of heading of approximately $\pm 180°$ is recorded. Thus the new recorded configuration of the robot is $\langle x + \epsilon_1, y + \epsilon_2, \theta \pm 180° + \epsilon_3\rangle$, where ϵ_i is the error due to inaccuracy in both measurement and movement. In earlier work [17], we treated all errors — in both location (x, y) and heading (θ) — as if they were normally distributed. However, the change in heading is different from changes in x and y, since angular measurements are *cyclic*. That is, a change in heading of $\theta°$ is the same as that of $\theta \pm 360°k$, for any integer k.

If we knew in advance, for every pair of states, the approximate change in heading $(\Delta\Theta)$ between them, we could have modeled it as normal with mean $\Delta\Theta$, and small variance σ^2. We could have adopted a convention, normalizing all angles to be within a cyclic range, e. g. $[-180°, 180°]$, (similarly we may use radians), and always chosen to take as the angular change between two points $min(|\Delta\theta|, 360° - |\Delta\theta|)$, and assigned it the correct sign. Such an approach of using a non-circular distribution is justified when the estimation of a position is based only on readings a-priory known to be taken near this position, (see for example work by Thrun et al [20] and Lu et al [12]). However, *we do not know in advance the angles between states.* The data is a sequence of measurements recorded at all the states. We *estimate* the probabilities of the states in which they were recorded, and take a *weighted mean* of the measurements in order to estimate the angular change between every two states. Thus, we are facing the following problem: *What is the interpretation of a "mean angle"?*

As an example, suppose we want to estimate the heading change from state a to state b of Figure 1. We adopt the convention of angles being expressed between $-180°$ and $180°$. Also, suppose that the robot recorded two measurements of angular distance from state a to state b: $-169°$ and $185°$. The simple average between these measurements is an estimate of the mean heading change of $8°$. Obviously this value does not even approximate the change of heading between the two states. The same problem arises if we use any other convention for expressing angles (e.g. $0°$ to $360°$). The problem lies in the fact that angles that are about $180°$ away from the mean angle, indeed greatly deviate from this mean, while angles that deviate about $360°$ are actually very close to it. To capture this idea, the concept of *circular distribution* is required. We provide a brief introduction to the concepts and techniques used for handling directional data. In particular we concentrate on the *von Mises distribution* — a circular version of the normal distribution. Further discussion can be found in the statistical literature [6, 10, 13]. Section 3.3 returns to show how the theory is applied in our model and learning algorithm.

3.1 STATISTICS OF DIRECTIONAL DATA

Directional data in the 2-dimensional space can be represented as a collection of 2-dimensional vectors, $(\langle x_1, y_1\rangle, \ldots \langle x_n, y_n\rangle)$, on the unit circle, as shown in Figure 2. The points can also be represented as the corresponding angles between the radii from the center of the unit circle and the x axis, $(\theta_1, \ldots, \theta_n)$, respectively. The relationship between the two representations is:

$$x_i = \cos(\theta_i), \qquad y_i = \sin(\theta_i)\,, (1 \le i \le n)\,.$$

The vector mean of the n points, $\langle \overline{x}, \overline{y}\rangle$, is calculated as:

$$\overline{x} = \frac{\sum_{i=1}^{n}\cos(\theta_i)}{n}, \qquad \overline{y} = \frac{\sum_{i=1}^{n}\sin(\theta_i)}{n}\,. \tag{1}$$

Using polar coordinates, we can express the mean vector in terms of angle, $\overline{\theta}$, and length, $\overline{a}$, where (except for the case $\overline{x} = \overline{y} = 0$):

$$\overline{\theta} = \arctan(\frac{\overline{y}}{\overline{x}}), \qquad \overline{a} = (\overline{x}^2 + \overline{y}^2)^{\frac{1}{2}} \tag{2}$$

The angle $\overline{\theta}$ is the mean angle, while the length $\overline{a}$ is a measure (between 0 and 1) of how concentrated the sample angles are around $\overline{\theta}$. The closer $\overline{a}$ is to 1, the more concentrated the sample is around the mean, which corresponds to a smaller sample variance.

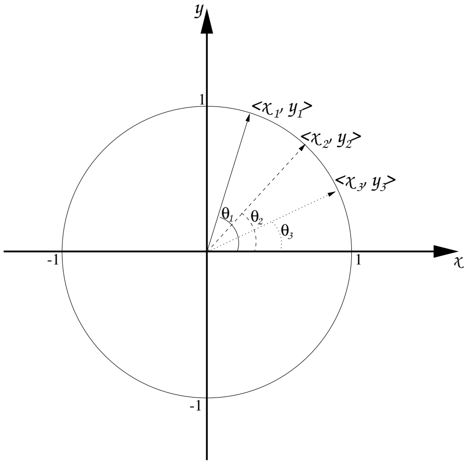

Figure 2: Directional data represented as angles and as vectors on the unit circle.

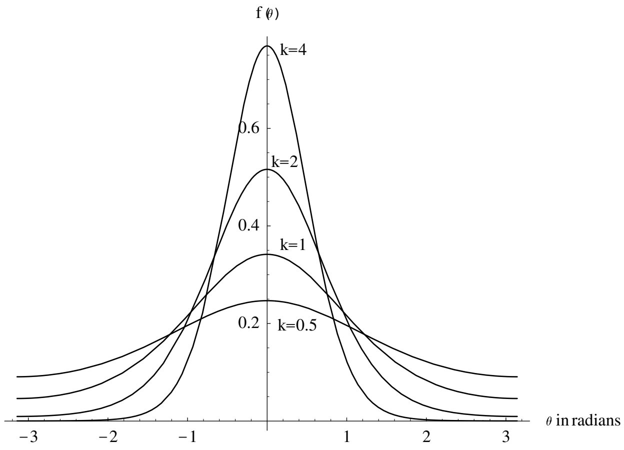

Figure 3: The von Mises distribution with mode 0 and various k values.

A function f is a density function of a continuous *circular distribution* if and only if: $f(x) \geq 0$ and $\int_0^{2\pi} f(x)dx = 1$. A simple example of a circular distribution is the *uniform circular distribution*, whose density function is $f(\theta) = \frac{1}{2\pi}$ (where θ is measured in radians).

One way of deriving a circular version of an unlimited linear distribution is through "wrapping" it around a circumference of the unit circle. If x is a random variable on the line with probability density function $f(x)$, the wrapped random variable $x_w = [x \bmod 2\pi]$ is distributed according to a wrapped distribution with the probability density function: $f_w(\theta) = \sum_{-\infty}^{\infty} f(\theta + 2\pi k)$. Applying this derivation to the normal distribution results in a circular version of the normal distribution, but estimating its parameters from sample data can be hard [6, 13]. An easier-to-estimate circular version of the normal distribution was derived, by von Mises [6, 13]. We use this distribution to model the robot heading in this work, and it is described below.

3.2 THE VON MISES DISTRIBUTION

A circular random variable, θ, $0 \leq \theta \leq 2\pi$, is said to have the *von Mises distribution* with parameters μ and k, where $0 \leq \mu \leq 2\pi$ and $k > 0$, if its probability density function is:

$$f_{\mu,k}(\theta) = \frac{1}{2\pi I_0(k)} e^{k\cos(\theta-\mu)} \quad ,$$

where $I_0(k)$ is the modified Bessel function of the first kind and order 0:

$$I_0(k) = \sum_{r=0}^{\infty} \frac{1}{r!^2} (\frac{1}{2}k)^{2r} \quad .$$

Similar to the linear normal distribution, this is a unimodal distribution, symmetrical around μ. The mode is at $\theta = \mu$ while the antimode is at $\theta = \mu + \pi$. We observe that the ratio of the density at the mode to the density at the antimode is e^{2k}, which indicates that the larger k is, the more concentrated the density is about the mode. Figure 3 shows an "unwrapped" plot of the von Mises distribution for various values of k where $\mu = 0$.

We now describe how to estimate the parameters μ and k given a set of heading samples (angles $\theta_1, \ldots \theta_n$) from a von Mises distribution [13]. We are looking for maximum likelihood estimates for μ and k. The likelihood function for the data generated by a von Mises distribution with parameters μ and k is:

$$L_{\mu,k} = \prod_{i=1}^{n} f_{\mu,k}(\theta_i) = \frac{e^{k\sum_{i=1}^{n}\cos(\theta_i-\mu)}}{(2\pi)^n I_0(k)^n} \quad .$$

The maximum likelihood estimate for μ, $\overline{\mu}$, is: $\overline{\mu} = \arctan(\frac{\overline{y}}{\overline{x}})$, where $\overline{y}$, $\overline{x}$ are as defined in equation 1.

The maximum likelihood estimate for k is the $\overline{k}$ that solves the equation:

$$\frac{I_1(\overline{k})}{I_0(\overline{k})} = \frac{1}{n}\sum_{i=1}^{n} \cos(\theta_i - \mu) \quad . \tag{3}$$

If we don't know μ and are only interested in estimating k with respect to the *estimate* $\overline{\mu}$, by using trigonometric manipulation and the definition of $\overline{a}$ (Equation 2), we can substitute the right hand side of equation 3 by $\overline{a}$ and obtain that the maximum likelihood estimate for k is $\overline{k}$ that satisfies: $\frac{I_1(\overline{k})}{I_0(\overline{k})} = \overline{a}$.

However, if we do have a given μ and want to find a maximum likelihood estimate for the concentration k of the sample data around that specified μ, we need to use as a maximum likelihood estimate for k, $\overline{k}$ that satisfies:

$$\frac{I_1(\overline{k})}{I_0(\overline{k})} = \frac{1}{n}\sqrt{\left(\sum_{i=1}^{n}\cos(\theta_i)\right)^2 + \left(\sum_{i=1}^{n}\sin(\theta_i)\right)^2 - \left(\sum_{i=1}^{n}\sin(\mu - \theta_i)\right)^2} \quad .$$

The above estimation formulae agree with the intuition that the sample is more concentrated ($\overline{k}$ is larger) about the sample mean ($\overline{\mu}$) than about the true distribution mean (μ).

The rest of the section explains how the von Mises parameters are incorporated into the Hidden Markov model, and how the learning algorithm is adapted to learn these parameters.

3.3 HANDLING ANGULAR ODOMETRIC READINGS

To model the heading difference between each pair of states, the relation matrix R, described in Section 2, is 3-dimensional, consisting of the components $\langle x,\ y,\ \theta\rangle$. The component $R_{i,j}[\theta]$ represents the heading change of moving from state s_i to s_j, and is assumed to be distributed according to the von Mises distribution. The notation $\mu^{\theta}_{i,j} \stackrel{\text{def}}{=} \mu(R_{i,j}[\theta])$ represents the mean of the distribution for this heading change, while $k^{\theta}_{i,j} \stackrel{\text{def}}{=} k(R_{i,j}[\theta])$ represents the concentration parameter around the mean[3]. The three constraints described before for the components of R, (ideally) hold for the θ component as well.

Similarly, every observed relation item, r_t, in the experience sequence E, has a heading-change component, θ, which records the robot's estimated change in heading between the state at time t, q_t, and the state q_{t+1}.

The reestimation formula for the von Mises mean parameter of the heading change between states s_i and s_j is:

$$\overline{\mu}^{\theta}_{i,j} = \arctan\left(\frac{\displaystyle\sum_{t=0}^{T-2}[\sin(r_t[\theta])\xi_t(i,j) - \sin(r_t[\theta])\xi_t(j,i)]}{\displaystyle\sum_{t=0}^{T-2}[\cos(r_t[\theta])\xi_t(i,j) + \cos(r_t[\theta])\xi_t(j,i)]}\right).$$

The fraction denotes the ratio between the expected sine and the expected cosine of the heading change from state i to state j. Since the heading change from j to i is identical in magnitude but opposite in direction to the heading change from i to j, the transitions from j to i are also accumulated – with reversed signs. By taking $arctan$ of this ratio we get an estimate for the mean heading change itself.

To reestimate the concentration parameter, we need to find $\overline{k}^{\theta}_{i,j}$ such that:

$$\frac{I_1[\overline{k}^{\theta}_{i,j}]}{I_0[\overline{k}^{\theta}_{i,j}]} = \frac{\sum_{t=0}^{T-2}[\xi_t(i,j)\cos(r_t[\theta] - \overline{\mu}^{\theta}_{i,j})]}{\sum_{t=0}^{T-2}\xi_t(i,j)} \ .$$

[3] In contrast, x and y are normally distributed and have their *variance* rather than *concentration* stored in R.

Finding $\overline{k}^{\theta}_{i,j}$ that satisfies this equation is done through the use of a lookup table listing values of the quotient $\frac{I_1[x]}{I_0[x]}$.

The above reestimation formulae agree with the maximum likelihood estimator formulae given in Section 3.1. Their correctness can be proved along the lines of the proof provided in our previous document [16].

4 STATE-RELATIVE COORDINATE SYSTEMS

In our previous work we assumed that there is a single global coordinate system within which the robot operates. Moreover, we assumed that the robot collects its data within a perpendicular corridor framework and that it takes advantage of this single perpendicular framework while recording odometric information. This assumption may be troublesome in practice. The rest of the paper discusses the potential problems, presents a method for relaxing the assumptions and addressing the problems, and demonstrates the effectiveness of the solutions through experiments and results.

4.1 MOTIVATION

We tend to think about an environment as consisting of landmarks fixed in a global coordinate system and corridors or transitions connecting these landmarks. However, this view may be problematic when robots are involved.

Conceptually, a robot has two levels in which it operates; the *abstract* level, in which it centers itself through corridors, follows walls and avoids obstacles, and the *physical* level in which motors turn the wheels as the robot moves. In the physical level many inaccuracies can occur: unaligned wheels or unsynchronized motors can cause sidewards drift, an obstacle under a wheel can cause the robot to slightly rotate around itself, or uneven floors may cause the robot to slip in a certain direction. In addition, the odometric measuring instrumentation may be inaccurate in and of itself. In the abstract level, corrective actions are constantly executed to overcome the physical drift and drag. For example, if the left wheel is disaligned and drags the robot leftwards, a corrective action of moving to the right is constantly taken in the higher level to keep the robot centered in the corridor.

Such phenomena greatly effect the odometry recorded by the robot, if it is interpreted with respect to one global framework. For example, consider the robot depicted in Figure 4. It drifts to the left $-\phi^{\circ}$ when moving from one state to the next, and corrects for it by moving ϕ° to the right to maintain itself centered in the corridor, moving along the solid arrow. Let us assume that states are lo-

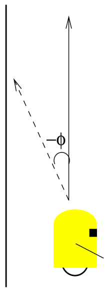

Figure 4: The robot moves in a corridor along the solid arrow, correcting for drift in the direction of the dashed arrow.

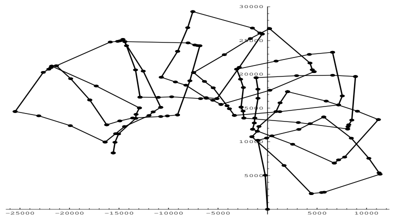

Figure 5: A path in a perpendicular environment, plotted based on odometric readings taken by the robot Ramona.

cated along the center of the corridor, which is aligned with the y axis of the global coordinate system. The robot steps back and forth in the corridor. Whenever it reaches a state, its odometry reading changes by $\langle x, y, \theta \rangle$ along the $\langle X, Y,$ heading $\rangle$ dimensions, respectively. As the robot proceeds, the deviation with respect to the x axis becomes more and more severe. Thus, after going through several transitions, the odometric changes recorded between every pair of states, with respect to a global coordinate system, become larger and larger (especially in the X dimension).

Similar problems of inconsistent odometric changes recorded between pairs of states can arise along any of the odometric dimensions. It is especially severe when such inconsistencies arise with respect to the heading, since this can lead to confusion between the X and the Y axes, as well as confusion between forwards and backwards movement (when the deviation in the heading is around $90°$ or $180°$ respectively). An example of our robot view of a perfectly perpendicular office environment, based on its odometric readings within a global coordinate system, is shown in Figure 5. The data was collected by our robot Ramona, while moving along the corridors in an area of our department, depicted in Figure 7.

A solution to such a situation is to model the odometric relations of moving from state s_i to state s_j using a changing coordinate system which is *respective* to state s_i, as opposed to a global coordinate system anchored at the initial state. We formalize this idea and provide the update rules for the odometric information based on this approach in the rest of this section. We have implemented our solution, and demonstrate its effectiveness throughout Section 5.

4.2 LEARNING ODOMETRIC RELATIONS WITH CHANGING COORDINATES

As before, our experience sequence E consists of T pairs $\langle r_t, V_t \rangle$ of recorded odometric relations and observation vectors. The odometric relations are still recorded with respect to the robot's global coordinate system. However, when learning the relation matrix from the odometric readings, we interpret the entry $R_{i,j}$ in the relation matrix R, as encoding the information with respect to a coordinate sys-

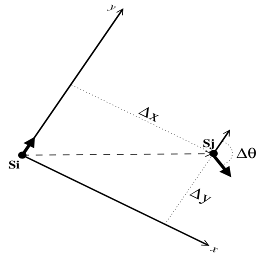

Figure 6: Robot in state S_i, facing in the direction of the y axis.

tem whose origin is anchored at the state s_i; the y axis is aligned with the robot's heading in state s_i and the x axis is perpendicular to it. This is depicted in figure 6. The robot is in state s_i facing in the direction pointed to by the y axis. Its relationship to the state s_j is described in terms of the coordinate system shown in the figure. Its heading in each state is denoted by the bold arrow.

To support this interpretation of the relation matrix we need to revisit the formulation of the geometrical-consistency constraints stated in Section 2, as well as the update formulae used when learning the model.

The consistency constraints have to reflect the coordinate system with respect to which the odometry is represented. Since the heading measurement is independent of any specific coordinate system, only the constraints over the x and y components of the odometric relation need to be redefined. We denote by $\mu^{\langle x,y \rangle}(a, b)$ the vector $\langle \mu(R_{a,b}[x]), \mu(R_{a,b}[y]) \rangle$. Let us define $\mathcal{T}_{ab}$ to be the transformation which maps an $\langle x_a, y_a \rangle$ pair represented with respect to the coordinate system of state a, to the same pair represented with respect to the coordinate system of state b, $\langle x_b, y_b \rangle$, (note that $\mathcal{T}_{ab} = \mathcal{T}_{ba}^{-1}$).

More explicitly, as before, let $\mu^{\theta}(a, b)$ be the mean change in heading from state a to state b (recall that $\mu^{\theta}(a, b) = -\mu^{\theta}(b, a)$). The transformation $\mathcal{T}_{ab}$ is defined as follows:

$$\begin{bmatrix} x_b \\ y_b \end{bmatrix} = \mathcal{T}_{ab}\left(\begin{bmatrix} x_a \\ y_a \end{bmatrix} \right) = \begin{bmatrix} x_a \cos(\mu^{\theta}(a,b)) - y_a \sin(\mu^{\theta}(a,b)) \\ x_a \sin(\mu^{\theta}(a,b)) + y_a \cos(\mu^{\theta}(a,b)) \end{bmatrix}.$$

We can now redefine the consistency constraints for the x and y components of the odometric relation:

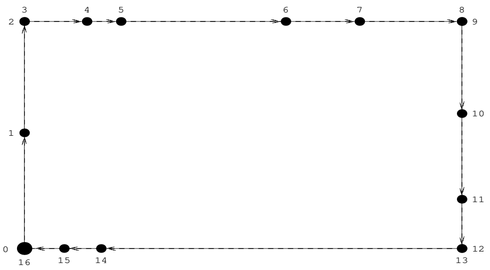

Figure 7: Model of a prescribed path through a true hallway environment.

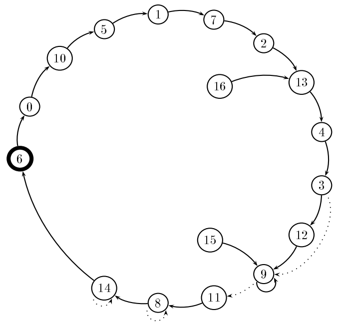

Figure 8: Learned topological model.

- $\diamond$ $\mu^{\langle x,y\rangle}(a,a) = \langle 0,0\rangle$;
- $\diamond$ $\mu^{\langle x,y\rangle}(a,b) = -\mathcal{T}_{ba}\left(\mu^{\langle x,y\rangle}(b,a)\right)$ *(anti-symmetry)*;
- $\diamond$ $\mu^{\langle x,y\rangle}(a,c) = \mu^{\langle x,y\rangle}(a,b) + \mathcal{T}_{ba}\left(\mu^{\langle x,y\rangle}(b,c)\right)$*(additivity)*;

The reestimation formulae for all the parameters except for the x and y components of the relation matrix R, remain as before. However, the reestimation formulae for the x and y parameters are changed to reflect the relative coordinate systems used. $\mu^x_{i,j}$ and $\mu^y_{i,j}$ are reestimated as follows:

$$\begin{bmatrix}\overline{\mu}^x_{i,j}\\ \overline{\mu}^y_{i,j}\end{bmatrix} = \frac{\displaystyle\sum_{t=0}^{T-2}\xi_t(i,j)\begin{bmatrix}r_t[x]\\ r_t[y]\end{bmatrix} - \sum_{t=0}^{T-2}\xi_t(j,i)\mathcal{T}_{ji}\left(\begin{bmatrix}r_t[x]\\ r_t[y]\end{bmatrix}\right)}{\displaystyle\sum_{t=0}^{T-2}\left(\xi_t(i,j)+\xi_t(j,i)\right)} \ .$$

These reestimation rules are guaranteed to satisfy the first two geometrical constraints, but not the additivity constraint. Their correctness can be proved along the lines of the correctness proofs for all other formulae [16].

5 EXPERIMENTS AND RESULTS

The goal of this work is to use odometry to improve the learning of topological models, while using fewer iterations and less data. We tested our algorithm in a simple robot-navigation world. In earlier stages of this work, a strong assumption underlay our experiments: the corridors in the environment are all perpendicular to each other, and the agent was using this perpendicularity to reset its position while accumulating the odometric readings. Here we have updated the algorithm and dropped the assumption. The experiments demonstrate that the use of odometry, even with accumulated rotational error and without using the perpendicularity assumption, is still very beneficial.

5.1 EXPERIMENTAL SETTING

Our experiments use both real robot data and simulated data. We ran our robot Ramona, a modified RWI B21, along a *prescribed*[4] directed path in our department corridors. Low-level routines let Ramona move forward through hallways from intersection to intersection and to turn ninety degrees to the left or right. Ultrasonic data interpretation let her perceive, in three directions – front, left and right – whether there is an open space, a door, a wall, or something unknown. Doors and intersections constitute *states*. When they are detected by Ramona, it stops and records its observations, as well as its odometric change between the previous and the current state. All recorded measures as well as the actions are, of course, subject to error.

The path Ramona followed consists of 4 connected corridors, which include 17 states, as shown in Figure 7. Black dots represent the physical locations of states. Multiple states (depicted as numbers in the plot) associated with a single location correspond to different orientations of the robot at that location. The larger black circle, at the bottom left corner, represents the starting position. The observations associated with each state are omitted for clarity. A projection of the odometric readings that Ramona recorded along the x and y dimensions, is shown in figure 5.

To statistically evaluate our algorithm, we use a simulated office environment in which the robot follows a prescribed path. It is represented as an HMM consisting of 44 states, and the associated transition, observation, and odometric distributions. Figure 9 depicts this HMM. Arrows represent transitions that have probability 0.2 or higher. Solid arrows represent the most likely transitions between the states. We generated 5 data sequences from the model, each of length 800, using Monte Carlo sampling. One of these sequences is depicted in Figure 10. Again, observations are omitted, and this is a projection of the odometry readings onto a global 2-dimensional coordinate system. For each sequence we ran our algorithm 10 times. We also ran the standard Baum-Welch algorithm, not using odometric information, 10 times on each sequence. For both algorithms we started each run from a randomly picked initial model.

5.2 RESULTS

We used our algorithm to learn a topological model of the environment from the data gathered by Ramona. Figure 8 shows the topology of one typical learned HMM. The bold circle represents the initial state. The arrows semantics is

[4] Hence, no decisions are executed by the robot, and the model is an HMM and not a complete POMDP.

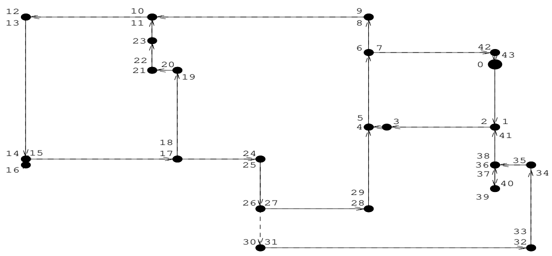

Figure 9: Model of a prescribed path through the simulated hallway environment.

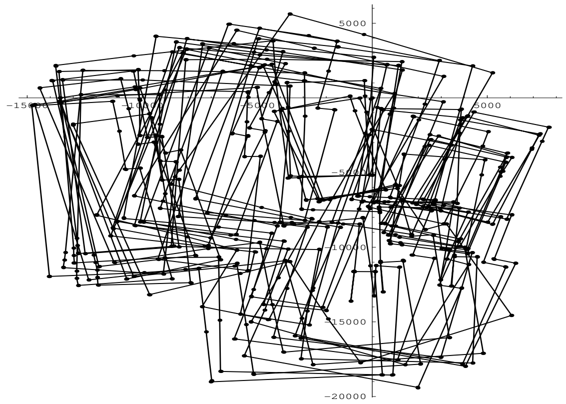

Figure 10: A data sequence generated by our simulator.

as stated before. It is clear that the learned topology corresponds well to the topology of the true environment. The observation distributions learned are omitted from the figure, but they too correspond well to the walls, doors and openings encountered along the path, while incorporating the identification error resulting from noisy sensors.

Traditionally, in simulation experiments, learned models are 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 [11] between the two distributions is a measure of how far the learned model is from the true model. We report our simulation results in terms of a sampled version of the KL divergence, as described by Juang and Rabiner [9]. It is based on generating sequences of sufficient length 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 purely topological models that are learned with and without odometry.

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 simulation sequences under the two learning settings, averaged over 10 runs per sequence.

The table demonstrates that the KL divergence with respect to the true model for models learned using odometry, is about *4-5 times smaller* than for models learned without odometric data. To check the significance of our results

Table 1: Average results of 2 learning settings with 5 training sequences.

Seq. #		1	2	3	4	5
With	KL	1.115	1.100	1.095	1.139	1.129
Odo	**Iter #**	69.7	81.8	84.3	52.4	112.9
No	KL	5.575	4.499	4.997	4.491	5.791
Odo	**Iter #**	120.4	107.5	116.2	113.3	120.6

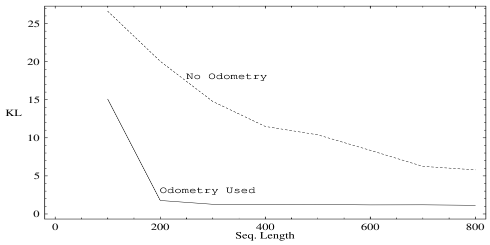

Figure 11: Average KL-divergence as a function of length.

we used the simple two-sample t-test. The models learned using odometric information have highly statistically significantly ($p \gg 0.9995$) lower average KL divergence than the others.

In addition, the number of iterations required for convergence when learning using odometric information is smaller than required when ignoring such information. Again, the t-test verifies the significance ($p > 0.995$) of this result.

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 800 (the complete sequence), in increments of 100, as individual sequences. We ran the two algorithmic settings over each of the 8 prefix sequences, 5 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 5 runs. Table 2 summarizes the results of this experiment. It lists the mean KL-divergence over the 5 runs for each of the prefixes, as well as the standard deviation around this mean. The plot in Figure 11 depicts the KL-divergence as a function of the sequence length for each of the settings. Both the table and the plot demonstrate that, in terms of the KL-divergence, our algorithm, which uses odometric information, is robust in the face of data reduction. In contrast, learning without the use of odometry is much more sensi-

Table 2: Average results with 8 incrementally longer sequences.

Seq. Length		800	700	600	500	400	300	200	100
With	**Mean** KL	1.136	1.201	1.191	1.241	1.216	1.272	1.771	15.076
Odo	**Std. Dev.**	0.091	0.083	0.131	0.082	0.036	0.085	0.510	12.884
No	**Mean** KL	5.790	6.249	8.354	10.390	11.490	14.772	20.044	26.619
Odo	**Std. Dev.**	0.554	0.937	0.179	0.460	0.422	1.280	0.904	0.460

tive to reduction in the amount of data. Again, we applied the two-sample t-test, which verified the statistical significance of these results.

6 CONCLUSIONS

Directional information which comes up in various applications of computer science in general and machine learning in particular, requires special treatment. Currently most statistical models and applications are based on distributions that are either discrete or continuous along the real line, rather than circular. It is important to be aware of the need for circular distributions as well as of their existence. Moreover, it would be useful to have widely used applications such as Autoclass [2] support such distributions.

A problematic aspect of directional data which manifests itself when learning maps and models for robot navigation is that of cumulative rotational errors. In the context of our work we have demonstrated that the use of relative coordinate systems rather than global ones supports learning relationship between states. The main point shown by this paper is that through correct treatment of directional data, odometric information which is weak and very noisy still provides a significant leverage when learning a purely topological map.

Acknowledgments

We thank Sebastian Thrun for his insightful comments, and Dimitris Michailidis for his editorial help. This work was supported by DARPA/Rome Labs Planning Initiative grant F30602-95-1-0020, by NSF grants IRI-9453383 and IRI-9312395, and by the Brown University Graduate Research Fellowship.

References

[1] A. R. Cassandra, L. P. Kaelbling, J. A. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 1996.

[2] P. Cheeseman, et al. Autoclass: A Bayesian classification system. In J. W. Shavlik, T. G. Dietterich, eds., *Readings in Machine Learning*. Morgan-Kaufmann, 1990.

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

[4] F. C. Dyer. Bees acquire route-based memories but not cognitive maps in a familiar landscape. *Animal Behaviour*, 41, 239–246, 1991.

[5] Z. Ghahramani, M. I. Jordan. Factorial hidden Markov models. In 15^{th} *Int. Conf. on Machine Learning*. 1997.

[6] E. G. Gumbel, J. A. Greenwood, D. Durand. The circular normal distribution: Theory and tables. *American Statistical Society Journal*, 48, 131–152, March 1953.

[7] D. Heckerman, D. Geiger. Learning Bayesian networks: A unification for discrete and Gaussian domains. In *Proc. of the* 11^{th} *Int. Conf. on Uncertainty in AI*. 1995.

[8] B. H. Juang. Maximum likelihood estimation for mixture multivariate stochastic observations of Markov chains. *AT&T Technical Journal*, 64(6), July-August 1985.

[9] B. H. Juang, L. R. Rabiner. A probabilistic distance measure for hidden Markov models. *AT&T Technical Journal*, 64(2), 391–408, February 1985.

[10] S. Kotz, N. L. Johnson, eds. *Encyclopedia of Statistical Sciences*, vol. 2, pp. 381–386. John Wiley and Sons, 1982.

[11] S. Kullback, R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1), 79–86, 1951.

[12] F. Lu, E. E. Millios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4, 333–349, 1997.

[13] K. V. Mardia. *Statistics of Directional Data*. Academic Press, 1972.

[14] I. Nourbakhsh, R. Powers, S. Birchfield. Dervish: An office-navigating robot. *AI Magazine*, 16(1), 53–60, 1995.

[15] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2), 257–285, February 1989.

[16] H. Shatkay, L. P. Kaelbling. Learning hidden Markov models with geometric information. Tech. Rep. CS-97-04, Dept. of Computer Science, Brown University, 1997.

[17] H. Shatkay, L. P. Kaelbling. Learning topological maps with weak local odometric information. In *Proc. of the* 15^{th} *Int. Joint Conf. on AI*. 1997.

[18] R. G. Simmons, S. Koenig. Probabilistic navigation in partially observable environments. In *Proc. of the Int. Joint Conf. on AI*. 1995.

[19] R. Smith, M. Self, P. Cheeseman. A stochastic map for uncertain spatial relationships. In S. S. Iyengar, A. Elfes, eds., *Autonomous Mobile Robots*. IEEE Press, 1991.

[20] S. Thrun, W. Burgard, D. Fox. A probabilistic approach to concurrent map acquisition and localization for mobile robots. *Machine Learning*, 31, 29–53, 1998.