

Optimizing Hidden Markov Models for Ocean Feature Detection

Sandeep Kumar

Indian Institute of Technology Madras
Chennai, India.

Sergio Jimenez Celorrio

Universidad Carlos III de Madrid
Leganes, Madrid, Spain

Frederic Py

Monterey Bay Aquarium Research Institute
Moss Landing, California, USA.

Deepak Khemani

Indian Institute of Technology Madras
Chennai, India.

Kanna Rajan

Monterey Bay Aquarium Research Institute
Moss Landing, California, USA.

Abstract

Given the diversity and spatio-temporal scales of dynamic coastal processes, sampling is a challenging task for oceanographers. To meet this challenge new robotic platforms such as Autonomous Underwater Vehicle (AUV) are being increasingly used. For effective water sampling during a mission an AUV should be adaptive to its environment, which requires it to be able to identify these dynamic and episodic ocean features *in-situ*. We describe the use of Hidden Markov Models (HMM) as a feature detection model used onboard an AUV, an autonomous untethered robot. We show how to build an identification model from data collected during past missions. Then we show how the parameters of the HMM can be optimized using a Genetic Algorithm approach, from models trained with the Baum-Welch algorithm in the initial population.

Introduction

The coastal ocean offers a range of biogeochemical phenomenon of interest to oceanographers. Often these phenomenon occur over unpredictable spatial and temporal extents and are difficult to observe using traditional ship-based oceanographic methods. To meet this challenge robotic platforms such as Autonomous Underwater Vehicles (AUVs), an untethered robotic platform, are being increasingly used as a cost effective way for sampling and observation. Robotic platforms such as AUVs are cost-effective, can be targeted towards specific features of scientific interest and are increasingly persistent in the water-column. Their effectiveness however, is often limited by their adaptability since the features in question are often dynamic and episodic and unpredictable in their spatial and temporal extents. To achieve its goals, the AUV must be able to identify whether or not it is within a feature of interest as detected by its sensors. Such identification allows the platform to use the paradigm of *sense-plan-act* to adapt its navigation and enable targeted water sampling for shore side analysis (McGann et al. 2008b; Py, Rajan, and McGann 2010).

One example of an AUV is Monterey Bay Aquarium Research Institute's (MBARI) Dorado platform. Fig. 1 shows an image of the AUV with its Gulper water sampling system (Bird, Sherman, and Ryan 2007). Based on sensed data

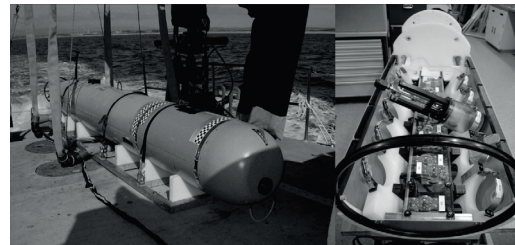


Figure 1: MBARI's Dorado AUV with a water sampler

the AUV can decide *in-situ* whether to trigger a sampler for water acquisition (McGann et al. 2008a). As the AUV is equipped with a small number of samplers (10 in our case) which can be actuated only once per mission sampling comes at a cost. To make an accurate decision the AUV should have precise information about the state of the feature of interest.

The *in-situ* identification of a given feature is defined as a classification task. Given an observation from the sensors, the aim is to determine the state of the feature of interest. One implication is to detect whether the vehicle is outside, on the boundary or inside some feature such as an INL (Intermediate Nepheloid Layer¹) or an algal bloom. To detect the state of a feature of interest for the AUV, we first need to build an identification model. Initial work in this area addressed the problem of *in-situ* identification of water-column features by standard classification (Fox et al. 2007). Such reactive models rely on immediate sensor reading, ignoring the past. More recent efforts in this direction involve generating a systematic identification model based on Hidden Markov Models (HMMs) (Rabiner 1989) for estimating the environments' state (Py, Celorrio, and Rajan 2010). HMMs exploit the sequential information of the sensor readings and thus provide more robust identification than standard classification.

In this paper we discuss the ocean feature-identification model for AUVs based on HMMs. This model is built using a two step machine learning process. First, semi-supervised clustering is done on the raw sensor data. The clusters thus identified are used, in the second step, as observations for

¹INLs are fluid sheets of suspended particulate matter that originate from the sea floor

building an HMM which serves as the identification model. The focus of our research here is to come up with the best possible parameter estimation scheme for the HMM. Traditional methods like the Baum-Welch (BW) algorithm (Baum et al. 1970) are unsupervised and offer locally optimal solutions. Furthermore they do not allow us to systematically bias the model easily.

We introduce labeled data and use Genetic Algorithms (GA) to further optimize the HMMs learned by the BW algorithm, aiming to produce models that better fit the data. GA also offers the flexibility to bias the model parameters allowing us to prevent false positive predictions for certain states.

This paper is organized as follows. A brief background on HMMs, their exploitation in-situ and the construction of the feature model is provided. We then discuss parameter optimization using GAs, the core of the paper. Experimental evaluation is followed by related work. The conclusion ends the paper.

Hidden Markov Model

A Hidden Markov Model is a state automaton in which the states are not directly visible. Instead we can access a set of observations which are probabilistically dependent on the current state of the system. For a HMM we need to define a discrete set of states $S = [s_1, s_2 \dots s_N]$ and a discrete set of observations $O = [o_1, o_2, \dots o_M]$.

A state at time t is denoted by q_t . A HMM uses the *Markov assumption* that the current state is only dependent on the previous state. Between each pair of states $i \in S$ and $j \in S$ a transition probability is defined as $a_{ij} = P(q_{t+1} = j | q_t = i)$

The transition probabilities among all the states are represented by the square matrix A whose each element is a_{ij} . Therefore we have the transition matrix:

$$A_{N,N} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,N} \end{pmatrix}$$

Any transition which does not exist in the model has a zero entry in the transition matrix. Every row of the transition matrix is subject to the constraint:

$$\sum_{j=1}^N a_{ij} = 1 \quad (1)$$

Each state also has a probability distribution over the possible observations that can be emitted. An observation at time t is denoted by o_t . Therefore, for each state $j \in S$, observation $k \in O$ pair, we have $b_j(k) = P(o_t = k | q_t = j)$ which gives the probability of observing the symbol k when in state j at time t . Hence we define the emission matrix B as follows:

$$B_{N,M} = \begin{pmatrix} b_1(1) & b_1(2) & \cdots & b_1(M) \\ b_2(1) & b_2(2) & \cdots & b_2(M) \\ \vdots & \vdots & \ddots & \vdots \\ b_N(1) & b_N(2) & \cdots & b_N(M) \end{pmatrix}$$

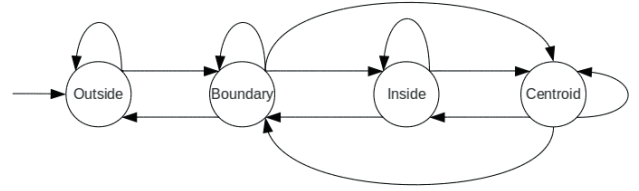


Figure 2: The HMM structure used in the identification model

where every row of the emission matrix is subject to the constraint:

$$\sum_{k=1}^M b_j(k) = 1 \quad (2)$$

The initial state probability vector which gives us the probability of starting in a particular state is defined as $\Pi = [\pi_1, \pi_2, \dots \pi_N]$ where each π_i gives the probability of starting in the state $i \in S$. Hence a complete HMM can now be defined by the parameter set:

$$\lambda = (A, B, \Pi)$$

Exploiting HMM for Feature Identification

HMMs are useful for classifying data which does not appear in isolation but are part of a time series. We use a HMM as an identification model for in-situ identification of ocean features during an AUV mission. The identification is based on the HMM and observed sensor values. More specifically we are interested in the probability of the feature being in a state j (for the purpose of robotic actuation) at the current time step t given an observation k and a model λ , i.e., $fwd(1 : t, j)$. This probability is recursively computed, taking into account historical observations as:

$$fwd(1 : t, j) = \alpha_t b_j(k) \sum_i a_{ij} fwd(1 : t - 1, i)$$

where α_t is a normalization factor such that $\sum_j fwd(1 : t, j) = 1$. The initial state of the HMM is Outside, therefore for the base case of $t = 1$, we have $\Pi = [Outside = 1, Inside = 0, Boundary = 0 \text{ and } Centroid = 0]$.

Building the Feature Identification Model

A two-step machine learning process is defined to automatically build the feature identification model for the AUV. The process takes as input the raw sensor data and gives as output a Hidden Markov Model. The output HMM serves as the identification model for the feature of interest. For example the HMM shown in Fig. 2 can identify four different states within the feature of interest. In this case the states *outside* the feature, at its *boundary*, *inside* or at the *centroid* are based on studies of INLs and their science requirements (McPhee-Shaw 2006; Ryan et al. 2010).

An overview of the process is shown in Fig 3. The sensor data available is partially labeled with the feature state it belongs to. Complete labeling of sensor data is usually not viable given the size of the datasets. For instance, a one

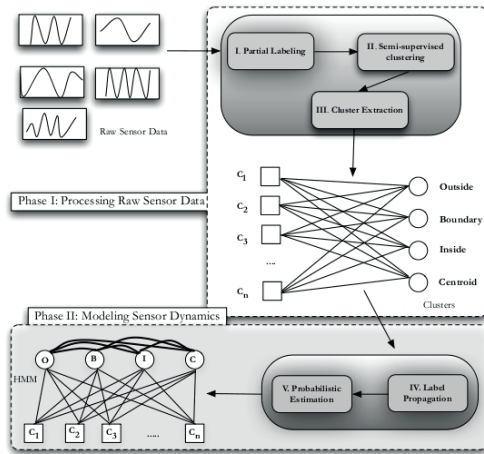


Figure 3: An overview of the process for building the identification model

hour AUV water-column survey with sensors sampling at 5 Hz generates approximately 15000 samples of raw sensor data. In the first phase semi-supervised clustering, using Self-Organizing Maps (SOM) (Kohonen 2001) is done on the raw sensor data which generates a set of clusters C^* that summarize the sensor readings. The clustering step allows the data vectors to be transformed into a discrete set of observations. Subsequently sensor data can be represented with the cluster id of the cluster it falls into. This gives us a dataset of discrete observations which are then used to build the HMM.

The second phase of building the feature identification model is started with the label propagation where unlabeled observations are labeled according to a prior probability distribution of clusters over feature states. Now with the fully labeled database and following frequency counting, a seed HMM is built. While building this seed model assumptions based on scientific knowledge of the features are made. For example as shown in Fig. 2 it is assumed that the first state will always be *outside* given that an AUV mission will often start well outside the boundaries of a feature.

In the next step the seed model is trained using the Baum-Welch (BW) algorithm (Baum et al. 1970; Rabiner 1989). It is a specific case of the Expectation-Maximization (EM) algorithm. Baum-Welch is an unsupervised training algorithm and takes unlabeled observations as input. Given a sequence of observations it finds the locally optimal set of parameters which maximize the probability of observing these sequences. As our optimization criteria is different and there is no guarantee of global optimal by Baum-Welch, we further optimize the learned model by using a Genetic Algorithm.

Parameter Optimization for HMM using GA

Genetic Algorithms (GA) (Srinivas and Patnaik 1994) are optimization algorithms that use biology-inspired mechanisms like mutation, crossover, natural selection, and survival of the fittest in order to refine a set of candidate solutions iteratively. Genetic Algorithms can be used to optimize

```

Initialize Population
Evaluate Population
repeat
  Select solutions for Next Population
  Perform Crossover and Mutation
  Evaluate Population
until Termination criterion reached

```

Figure 4: A simple Genetic Algorithm

for a wide variety of problems.

A genetic algorithm is usually given as shown in Fig. 4; we note that to use a GA we need an initial population of solutions, an evaluation or fitness function, a selection mechanism for selecting solutions for crossover, a crossover function which keeps the solutions consistent and a way of mutating the solutions. For optimizing the HMM parameters we encode them as follows:

Initial Population

We initialize the population with the seed HMMs and the HMMs learned using Baum-Welch. Each HMM is represented in terms of its parameter set $\lambda = (A, B, \Pi)$. In our case the starting state is always *outside*, hence Π does not have any significance. Thus we are left with the matrices A and B . Transition Matrix A is of size 4×4 as we have four states in our HMM. The emission Matrix B is of size 4×87 as the number of found clusters (observations) is 87. These clusters are found using semi-supervised SOM clustering.

Evaluation and Selection

Most of other algorithms for training HMMs like the Baum-Welch, are designed specifically to maximize the log probability of observing the training observation sequence, while GAs offer the flexibility to define an arbitrary evaluation function for which the HMM can be optimized. This is useful to systematically bias the HMM for preventing false positive predictions for certain states. Our AUV takes water samples at the *boundary* and *centroid* states, hence we want to prevent false predictions for these states which are temporally transitory.

Performance evaluation of a model is done using a confusion matrix (Provost, Fawcett, and Kohavi 1998) which allows us to see if the model is confusing between two states and hence giving us a quantitative measure of error. Each row in this matrix indicates actual state instances and each column represents the state predicted by the identification model. For all the identification models the predicted state is the one with the highest probability of occurrence with a good margin. The additional state *unknown* is predicted otherwise when states cannot be easily disambiguated. Each row of the confusion matrix is first normalized such that it sums up to 100. As a large percentage of our data belongs to the *outside* state, this avoids any bias towards a state which may contain more data points, by giving each state an equal weight.

To compute the final evaluation of a model we initially take the sum of the diagonal values of the confusion matrix.

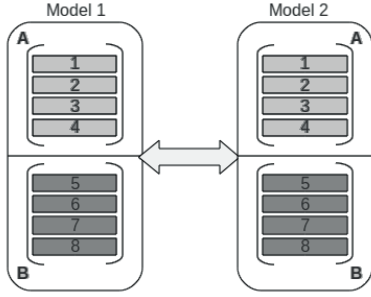


Figure 5: Crossover between two models

Values in the diagonal of the matrix indicate the proportion of predictions that were correct. From this we subtract the entries corresponding to false positives for *boundary* and *centroid* states. For a model λ with N states we have a $N \times N$ confusion matrix C with elements c_{ij} . The fitness of the model is calculated as follows:

$$Fitness(\lambda) = \sum_i c_{ii}^\lambda - \sum_{i \neq k_1} c_{ik_1}^\lambda - \sum_{i \neq k_2} c_{ik_2}^\lambda$$

where k_1 and k_2 are columns of C corresponding to the *boundary* and *centroid* states respectively.

Two models from the population are randomly selected for crossover at every cycle of the GA since the models with best fitness tend to have similar parameters.

Crossover

We are working with a non-random population of solutions (HMM parameters) which are subject to constraints in Eqns. 1 and 2. Therefore we only interchange the corresponding rows of the parameter matrices in a crossover. This helps in maintaining the consistency of solutions. For two models λ_1 and λ_2 we define a Transition Crossover i w.r.t. to the Transition Matrix A as follows:

$$a_{ij}^{\lambda_1} \longleftrightarrow a_{ij}^{\lambda_2}, \quad \forall j : 1 \leq j \leq N \quad (3)$$

Each $a_{ij}^{\lambda_1} \longleftrightarrow a_{ij}^{\lambda_2}$ symbolizes the swap of the corresponding entries of matrices A^{λ_1} and A^{λ_2} . A transition crossover shown in Eqn. 3 is a complete exchange of row i among the transition matrices of models λ_1 and λ_2 . There are N such transition crossovers possible. They are numbered from 1 to N , corresponding to each row i of transition matrix, for $1 \leq i \leq N$.

Similarly w.r.t. the Emission Matrix B we define the Emission Crossover $N + j$ as:

$$b_j^{\lambda_1}(k) \longleftrightarrow b_j^{\lambda_2}(k), \quad \forall k : 1 \leq k \leq M \quad (4)$$

Each $b_j^{\lambda_1}(k) \longleftrightarrow b_j^{\lambda_2}(k)$ symbolizes the swap of the corresponding entries of matrices B^{λ_1} and B^{λ_2} . An emission crossover shown in Eqn. 4 is a complete exchange of row j among the emission matrices of models λ_1 and λ_2 . There are N such emission crossovers possible. They are numbered from $N + 1$ to $2N$, corresponding to each row j of emission matrix, for $1 \leq j \leq N$.

Consider the case in Fig. 5. Numbers 1 to 4 symbolizes the four rows of the transition matrix in a model, and numbers 5 to 8 symbolizes the rows of the emission matrix. Given the model constraints, we can only exchange rows with the same number between the Model 1 and Model 2 for a valid crossover. An transition crossover would be a crossover between rows 1 to 4 and a emission crossover would be a crossover between rows 5 to 8. From a GA point of view there is no difference between the Transition Crossover and the Emission Crossover. We simply perform a random number of Transition Crossover and/or Emission Crossover in each cycle of the GA.

Mutation

Mutation is done by running a model through one iteration of the Baum-Welch algorithm. The seed model for this iteration is the current individual. The model resulting from the iteration of the Baum-Welch algorithm is the mutated model. Solutions produced by the mutation step are added back to the population.

The GA loop is terminated when no better individual is found for a constant number of iterations. The best performing individual is selected as the final solution.

Experimental Evaluation

The data used for the experiments focuses on the detection of INL states from different AUV missions. The data has been divided into two classes; learning missions and test missions. Table 1 highlights the dataset. The learning mission data consists of a set of approximately 500,000 observations from four different AUV missions between the years 2003 and 2005 sparsely labeled at between 7% and 9.7%. This data is used by the Baum-Welch algorithm in mutation and while generating a model for the initial population. Baum-Welch algorithm ignores the labels in the data. The testing data used for model evaluation consists of approximately 150,000 observations from thirteen missions between 2008 and 2010. As the evaluation makes use of labels, more test data is labeled at $\sim 21\%$.

Our experiments compare the performance of HMM models in terms of fitness over test missions. A detailed evaluation is done by using the normalized confusion matrix, used for calculating the fitness of a model. We select models at three different levels of optimization.

Boot-strap Model This model is constructed using learning data and serves as a seed model for Baum-Welch. First all unlabeled data points are labeled using a known probability distribution of observations over states. Then model parameters are estimated using a frequency counting technique. Table 2 shows the confusion matrix and fitness score for this model. The fitness value is 134.3 and the classification accuracy for *boundary* state is very poor.

Baum-Welch Model This model is built after running Baum-Welch over the boot-strap model. The learning data are used as a training set, after ignoring the labels.

Table 1: List of AUV missions used for experimentation

Mission # (yr-day)	Date	Labeled
Learning missions		
2003-308	Nov 4, 2003	7.0%
2004-028	Jan 28, 2004	7.8%
2004-321	Nov 16, 2004	7.1%
2005-306	Nov 2, 2005	9.7%
Test missions		
2008-315	Nov 10, 2008	13.0%
2008-318-1	Nov 13, 2008	16.3%
2008-318-2	Nov 13, 2008	9.8%
2009-308	Nov 4, 2009	12.9%
2009-313	Nov 9, 2009	25.0%
2009-342	Dec 8, 2009	35.0%
2009-348	Dec 14, 2009	22.7%
2010-081	Mar 22, 2010	10.6%
2010-082	Mar 23, 2010	23.8%
2010-083-1	Mar 24, 2010	27.4%
2010-083-2	Mar 24, 2010	29.7%
2010-118	Apr 28, 2010	28.3%
2010-119	Apr 29, 2010	22.7%

Table 2: Boot Strap Hidden Markov Model

	Out.	Bound.	In.	Cent.	Unknown
Out.	93.5	1.5	2.7	1.9	0.4
Bound.	66.4	2.6	9.2	0.0	21.7
In.	10.1	12.9	41.5	9.4	26.11
Cent.	17.3	11.1	33.2	33.5	4.9

Fitness Score = 134.3

Table 3: Baum-Welch Hidden Markov Model

	Out.	Bound.	In.	Cent.	Unknown
Out.	73.4	21.2	2.1	3.2	0.2
Bound.	23.6	54.0	6.6	0.0	15.8
In.	5.0	4.1	25.8	44.5	20.6
Cent.	5.2	1.6	18.4	71.0	3.8

Fitness Score = 149.6

The confusion matrix for this model is shown in the Table 3. Fitness for the model is 149.6 and the classification accuracy is 25.8 for the *inside* state. There are also large values of false positives for *boundary* and *centroid* states.

Genetic Algorithm Model This model is learned by using a GA with Boot-strap and the Baum-Welch models as the initial population. The confusion matrix for this model is shown in Table 4. The fitness for this model is 199.4. The percentage of false positives for the *boundary* is 5.0, 0.8 and 0.0 and for a *centroid* at 1.8, 0.0 and 8.8. This is low

Table 4: GA Hidden Markov Model

	Out.	Bound.	In.	Cent.	Unknown
Out.	89.0	5.0	3.7	1.8	0.4
Bound.	45.8	23.9	8.8	0.0	21.4
In.	3.6	0.8	63.4	8.8	23.3
Cent.	3.0	0.0	52.8	39.6	4.5

Fitness Score = 199.4

when compared to other models.

Related Work

HMMs are a common approach for handling sequential data with various applications in the field of speech processing and analyzing biological data. The idea of using Genetic Algorithms (GA) to learn HMMs has been used earlier. These attempts can be categorized into three classes (Volkert 2006).

Model parameter evolution Earlier work of Chau et al. (Chau et al. 1997) using GA for optimizing HMM parameters started with a random population of solutions. Their objective was to maximize the log probability of observing the training sequences, $\log P(O|\lambda)$. Experiments showed improved performance over the Baum-Welch algorithm.

Slimane et al. (Slimane et al. 1996) conducted a series of experiments to study the use of genetic algorithm for optimizing HMM parameters. Their findings suggested that HMM trained by Baum-Welch using seeds generated by GA gives the best quality models. Kwong et al. (Kwong and Chau 1997) optimized HMM parameters for speech recognition using GAs.

Evolution of model parameters and structure Kwong et al. (Kwong et al. 2001) also developed a hybrid-GA to evolve both the topology and the model parameters of a left-right HMM for speech recognition.

A genetic algorithm training approach combined with Baum-Welch has been used by Yada et.al to train HMM models representing signal patterns in DNA sequences (Yada et al. 1994; Yada 1996). It evolved both the topology and model parameters.

HMMs evolved both for topology as well as model structure have been used by Thomsen (Thomsen 2002) for solving the problem of predicting the secondary structure of protein sequences.

Model structure evolution Recent work on optimizing the structure of HMM for biological sequence analysis and secondary structure prediction also use the GA approach. Won et al. (Won, Prgel-Bennett, and Krogh 2004; Won et al. 2005) have focused their efforts on evolving only topology, while the task of learning parameters is left to the Baum-Welch algorithm.

Our work is within the first class, most closely related to the approach of Slimane et al. We also use a combination of Baum-Welch and Genetic Algorithm to optimize the HMM parameters. Our approach differs from the above, as we try

to optimize the model found by Baum-Welch using GA instead of using it for producing seed models for Baum-Welch. We do this because, first we have partially labelled data for producing good seed models for Baum-Welch. And second, because our evaluation function is also different. Given that in our work data examples are typically unbalanced, our evaluation function aims to maximize the normalized accuracy taking into account the cost of false positives for certain states, whereas most other algorithms try to optimize log probability of observing the training sequences $\log P(O|\lambda)$.

Conclusion

This paper presents an approach for in-situ identification of dynamic ocean features using a Hidden Markov Model. We present the use of Genetic Algorithms to further optimize the solution found by the Baum-Welch algorithm. Tables 3 and 4 show that more refined models can be found by using GAs over already optimized models. Additional optimization is not the only benefit of using GAs; the flexibility to design an arbitrary evaluation function allows introduction of systematic bias into the model where necessary. This is helpful in preventing false positives for certain states important in scenarios such as ours where the AUV has limited sample acquisition capability.

Acknowledgments

We thank the David and Lucile Packard Foundation, MBARI, the Univ. Carlos III and IIT Madras, for supporting our work.

References

Baum, L. E.; Petrie, T.; Soules, G.; and Weiss, N. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Annals of Mathematical Statistics* 41(1):164–171.

Bird, L.; Sherman, A.; and Ryan, J. P. 2007. Development of an Active, Large Volume, Discrete Seawater Sampler for Autonomous Underwater Vehicles. In *Proc Oceans MTS/IEEE Conference*.

Chau, C.; Kwong, S.; Diu, C.; and Fahrner, W. 1997. Optimization of hmm by a genetic algorithm. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97)*, volume 3, 1727–1730 vol.3.

Fox, M.; Long, D.; Py, F.; Rajan, K.; and Ryan, J. P. 2007. In situ analysis for intelligent control. In *Proc. of IEEE/OES OCEANS Conference*. IEEE.

Kohonen, T. 2001. *Self-Organisation Maps*, volume 30. Springer Series in Information Sciences.

Kwong, S., and Chau, C. 1997. Analysis of parallel genetic algorithms on HMM based speech recognition system. *IEEE Transactions on Consumer Electronics*, 43(4):1229–1233.

Kwong, S.; Chau, C. W.; Man, K. F.; and Tang, K. S. 2001. Optimisation of hmm topology and its model parameters by genetic algorithms. *Pattern Recognition* 34(2):509–522.

McGann, C.; Py, F.; Rajan, K.; Ryan, J. P.; and Henthorn, R. 2008a. Adaptive Control for Autonomous Underwater Vehicles. In *AAAI*.

McGann, C.; Py, F.; Rajan, K.; Thomas, H.; Henthorn, R.; and McEwen, R. 2008b. A Deliberative Architecture for AUV Control. In *ICRA*.

McPhee-Shaw, E. 2006. Boundary-interior exchange: Re-viewing the idea that internal-wave mixing enhances lateral dispersal near continental margins. *Deep Sea Research Part II: Topical Studies in Oceanography* 53(1-2):42–59.

Provost, F. J.; Fawcett, T.; and Kohavi, R. 1998. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98*, 445–453. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Py, F.; Celorrio, S.; and Rajan, K. 2010. Modeling dynamic coastal ocean features for in-situ identification and adaptive sampling. *Journal of Atmospheric and Ocean Technology - Oceans*. Submitted, in Review.

Py, F.; Rajan, K.; and McGann, C. 2010. A Systematic Agent Framework for Situated Autonomous Systems. In *9th AAMAS*.

Rabiner, L. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2):257–286.

Ryan, J. P.; Johnson, S.; Sherman, A.; Rajan, K.; Py, F.; Paduan, J.; and Vrijenhoek, R. 2010. Intermediate nepheloid layers as conduits of larval transport. *Limnology & Oceanography Methods* 8:394–402.

Slimane, M.; Venturini, G.; Asselin de Beauville, J.; Brouard, T.; and Brandeau, A. 1996. Optimizing hidden markov models with a genetic algorithm. In *Artificial Evolution*, volume 1063 of *LNCS*. Springer Berlin / Heidelberg, 384–396.

Srinivas, M., and Patnaik, L. 1994. Genetic algorithms: a survey. *Computer* 27(6):17–26.

Thomsen, R. 2002. Evolving the topology of hidden markov models using evolutionary algorithms. In *Parallel Problem Solving from Nature PPSN VII*, volume 2439 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 861–870.

Volkert, L. 2006. Investigating ea based training of hmm using a sequential parameter optimization approach. In *IEEE Congress on Evolutionary Computation, 2006. CEC 2006.*, 2742–2749.

Won, K. J.; Hamelryck, T.; Prugel-Bennett, A.; and Krogh, A. 2005. Evolving hidden markov models for protein secondary structure prediction. In *IEEE Congress on Evolutionary Computation*, 33–40.

Won, K. J.; Prugel-Bennett, A.; and Krogh, A. 2004. Training HMM structure with genetic algorithm for biological sequence analysis. *Bioinformatics* 20(18):3613–3619.

Yada, T.; Ishikawa, M.; Tanaka, H.; and Asai, K. 1994. Dna sequence analysis using hidden markov model and genetic algorithm. *Genome Informatics* 5:178–179.

Yada, T. 1996. Extraction of hidden markov model representations of signal patterns in dna sequences. *Pacific Symposium on BioComputing, Hawaii, USA*.