Semi-Supervised Learning of Hidden Markov Models via a Homotopy Method

Shihao Ji, Layne T. Watson, and Lawrence Carin

Abstract

Hidden Markov model (HMM) classifier design is considered for analysis of sequential data, incorporating both labeled and unlabeled data for training; the balance between the use of labeled and unlabeled data is controlled by an allocation parameter $\lambda \in [0, 1)$, where $\lambda = 0$ corresponds to purely supervised HMM learning (based only on the labeled data) and $\lambda = 1$ corresponds to unsupervised HMM-based clustering (based only on the unlabeled data). The associated estimation problem can typically be reduced to solving a set of fixed point equations in the form of a "natural-parameter homotopy". This paper applies a homotopy method to track a continuous path of solutions, starting from a local supervised solution ($\lambda = 0$) to a local unsupervised solution ($\lambda = 1$). The homotopy method is guaranteed to track with probability one from $\lambda = 0$ to $\lambda = 1$ if the $\lambda = 0$ solution is unique; this condition is not satisfied for the HMM, since the maximum likelihood supervised solution ($\lambda = 0$) is characterized by many local optima. A modified form of the homotopy map for HMMs assures a track from $\lambda = 0$ to $\lambda = 1$. Following this track leads to a formulation for selecting $\lambda \in [0, 1)$ for a semi-supervised solution, and it also provides a tool for selection from among multiple local-optimal supervised solutions. The results of applying the proposed method to measured and synthetic sequential data verify its robustness and feasibility compared to the conventional EM approach for semi-supervised HMM training.

Index Terms

Semi-supervised learning, homotopy method, hidden Markov models (HMMs), supervised learning.

S. Ji and L. Carin are with the Department of Electrical and Computer Engineering, Duke University, Box 90291, Durham, NC 27708-0291; L. T. Watson is with the Departments of Computer Science and Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0106.

I. INTRODUCTION

A classifier is typically trained on data pairs, defined by feature vectors and corresponding class labels. Such a framework is called supervised learning. In most cases class labels are manually assigned by experts. It is therefore often expensive and time consuming to collect large quantities of labeled data. Because of this labeling cost, labeled data are often scarce in practice. Using limited labeled data, a classifier designed with supervised learning is often unreliable, manifesting poor generalization performance [1].

To overcome this problem a new technique, termed semi-supervised learning, has been proposed, in which unlabeled data (for which only the feature vectors are available) are integrated with labeled data when performing classifier design. Because unlabeled data can be collected easily, without labeling costs, semi-supervised learning has attracted interest in various applications, for data defined by single feature vectors [2]–[6] and for sequential data defined by a sequence of feature vectors [7].

This paper focuses on sequential data, modeled via hidden Markov models (HMMs) [8]. HMMs have underlying (hidden) states, with the sequence of states characterized via a Markov process. These models have been used extensively in two application areas. The first application is concerned with classification of sequential data in speech recognition [8], target classification [9], and computational biology [10], etc. In these tasks, given a sequence of data, HMMs are used to assign a class label to the entire sequence. The second application deals with estimating the underlying Markovian state sequence given observed sequential data. Examples of this application include part-of-speech tagging in natural language processing [11] and named-entity extraction in information extraction [12]. This paper concentrates on the first application, corresponding to classification of sequential data, with an HMM classifier designed by exploiting both labeled and unlabeled data sequences. Labeled data correspond to sequences for which the corresponding target classes are known, and unlabeled data correspond to sequential data without the corresponding labels.

A conventional semi-supervised approach for training a generative model (e.g., HMMs) is the expectation maximization (EM) algorithm [3], [13]. In this procedure, the missing labels of the

unlabeled data are treated as hidden variables and the optimality criterion is to maximize the joint likelihood of both labeled data and unlabeled data. There has been significant previous work on semi-supervised learning, with notable successes; however, there also exist practical examples of performance degradation with the EM approach for semi-supervised learning. For example, Shahshahani *et al.* [4] describe degradation in image understanding by using Gaussian mixture models (GMM), while Nigam *et al.* [5] report degradation in naive Bayes classifiers for text classification, and Inoue *et al.* [7] observe degradation in HMM classifiers. Performance degradation has been principally attributed to model deviations and numerical instability [4]–[6]. Focusing on the former issue, Cozman *et al.* [6] analyze the degradation by examining asymptotic behavior of the maximum likelihood estimator. They state that when the model fits the data incorporating the unlabeled data reduces classification error, while when the model is incorrect the role of unlabeled data becomes complex and often results in degradation. On the other hand, Nigam *et al.* [5] speculate numerical problems in the EM algorithm, and suggest reducing the degradation by weighting the contribution from the unlabeled data, using methods such as the so-called EM-λ algorithm. However, the choice of suitable scalar parameter λ remains an important issue.

Our approach principally follows the previous work of Corduneanu and Jaakkola [14], who used the homotopy method [15]–[17] for semi-supervised learning, for choosing the parameter $\lambda \in [0, 1)$ that yields a proper balance between the use of labeled and unlabeled data. In this method semi-supervised learning is regarded as a problem of balancing heterogeneous data sources, with the allocation $\lambda \in [0, 1)$ representing the relative balance of two data sources, where $\lambda = 0$ corresponds to purely supervised learning, and $\lambda = 1$ to purely unsupervised. The proper balance λ is sought as the algorithm gradually morphs the supervised learning problem ($\lambda = 0$) into an unsupervised learning problem ($\lambda = 1$). The associated estimation problem can typically be reduced to solving a set of fixed point equations in the form of a "natural-parameter homotopy" [16]. Obviously, the EM algorithm may be used to compute solutions at an increasing sequence of values $\lambda \in [0, 1)$. However, it remains unclear how to choose the step size for increasing λ, this EM-based path following may fail if the path of solutions has turning points, and more importantly, the discrete path of EM-solutions, each of which is optimized independently, does not provide an indication of which λ is optimal for semi-supervised learning. These difficulties

of the EM-based path following are addressed by the homotopy method [16], which tracks a smooth nonbifurcating path of solutions by following increasing arc length along the solution path with an automatically determined optimal step size. Note that the EM-based continuation method assumes λ monotonically increases along the solution path, whereas a homotopy method permits λ to both increase and decrease along the solution path.

Being the first to extend this technique to semi-supervised-learning problems, Corduneanu and Jaakkola [14] applied the homotopy method on relatively simple graphical models, such as naive Bayes and Gaussian mixture models (GMM). Using these models, the supervised learning problem ($\lambda = 0$) has a unique solution, and (making a reasonable transversality assumption) the theory of globally convergent homotopy algorithms [15], [16] provides a strong existence guarantee of a unique smooth nonbifurcating path of fixed points. However, for the case of more general graphical models (e.g., HMMs), the supervised solution is no longer unique and is rather characterized by multiple local optima. We show that the theory of globally convergent probability-one homotopy algorithms can be tailored to this more general setting, and a smooth nonbifurcating path of fixed points from $\lambda = 0$ to $\lambda = 1$ can be identified. We also demonstrate that this framework allows a means of using the unlabeled data to select from among the multiple ($\lambda = 0$) supervised solutions.

The remainder of the paper is organized as follows. Section II presents a brief introduction to the homotopy method, along with three globally convergent probability-one homotopy maps for the problem of interest here. Section III formulates the optimality criterion of semi-supervised learning for generative models from a mutual information perspective, from which the source balancing problem is introduced. With this background, Sec. IV details the homotopy method for semi-supervised HMM training, with a maximum entropy criterion for choosing the proper λ, as well as how to use unlabeled data to select a preferential local supervised solution. Experimental results on measured and synthetic data are provided in Sec. V, followed in Sec. VI by conclusions and a discussion of future work.

II. Globally Convergent Probability-One Homotopy Method

The theory of globally convergent probability-one homotopy maps concerns finding zeros or fixed points of nonlinear systems of equations [15], [16]. The underlying idea is simple: Given a twice continuously differentiable function $F : \mathbb{R}^n \to \mathbb{R}^n$ of which a zero is sought, rather than solving the original difficult problem $F(z) = 0$ directly, start from an "easy" problem $G(z) = 0$ whose solution is readily identified, and gradually transform the "easy" problem into the original one, tracking the solutions along the transformation. Typically, one may choose a convex homotopy map, such as

$$H(\lambda, z) = (1 - \lambda)G(z) + \lambda F(z), \tag{1}$$

and trace an implicitly defined zero curve[1] $\gamma \in H^{-1}(0)$ from a starting point $(0, z)$ to a final point $(1, \bar{z})$. If this succeeds, then a zero point $\bar{z}$ of F is obtained.

Generally, there are two issues with respect to the homotopy method: (i) whether we can be assured that there exists a smooth path of solutions starting from $\lambda = 0$ and reaching a target solution at $\lambda = 1$ in finite arc length, and (ii) development of numerical techniques for tracing this path. In the following, we discuss three special homotopy maps that assure the properties desired in (i), with probability one; issue (ii) is also discussed below.

We first consider the globally convergent probability-one fixed point homotopy map:

$$H(\lambda, z) = (1 - \lambda)(z - a) + \lambda(z - f(z)), \tag{2}$$

where $a \in \mathbb{R}^n$ is a constant vector. Under a mild condition from the following theorem [16], the existence of a smooth path of solutions reaching a fixed point $\bar{z}$ of f is almost always guaranteed (i.e., with probability one).

Theorem 1: Suppose that $B \subset \mathbb{R}^n$ is a compact, convex subset, and $f : B \to B$ is twice continuously differentiable. Then for almost all vectors $a \in \operatorname{int} B$, there is a zero path γ of H emanating from $(0, a)$, along which the $n \times (n+1)$ Jacobian matrix $DH(\lambda, z)$ has full rank, that does not intersect itself and is disjoint from any other zeros of H, and reaches an accumulation

[1]The symbol $H^{-1}(0)$ denotes the set of points (λ, z) that satisfy $H(\lambda, z) = 0$.

point $(1, \bar{z})$ for which $f(\bar{z}) = \bar{z}$. Furthermore, if the Jacobian matrix $DH(1, \bar{z})$ is nonsingular, then the zero path γ between $(0, a)$ and $(1, \bar{z})$ has finite arc length.

The above theorem requires that $a \in \operatorname{int} B$ be a constant vector. In order to deal with a more general case in which a is a function of b and z, i.e. $a(b, z)$, we introduce the second homotopy map:

$$H(b, \lambda, z) = (1 - \lambda)(z - a(b, z)) + \lambda(z - f(z)), \tag{3}$$

where the parameter vector b is crucial for the probability-one homotopy theory, as shown in the following theorem.

Theorem 2: Suppose that $B \subset \mathbb{R}^n$ is a compact, convex subset, and $U \subset \mathbb{R}^m$ is a nonempty open set. Let $f : B \to B$ and $a : U \times B \to \operatorname{int} B$ be C^2 maps, and assume that rank $D_b a(b, z) = n$ for all $(b, z) \in U \times B$. Further assume that for each $b \in U$, $a_b(z) = a(b, z)$ has a unique fixed point z_b at which rank $(I - Da_b(z_b)) = n$. Define $H : U \times [0, 1) \times B \to \mathbb{R}^n$ by

$$H(b, \lambda, z) = (1 - \lambda)(z - a(b, z)) + \lambda(z - f(z))$$

and define $H_b(\lambda, z) = H(b, \lambda, z)$. Then for almost all $b \in U$ there is a zero curve γ of $H_b(\lambda, z)$ emanating from $(0, z_b)$, along which the Jacobian matrix $DH_b(\lambda, z)$ has full rank, that does not intersect itself or other zeros of $H_b(\lambda, z)$, and reaches (accumulates at) a fixed point $\bar{z}$ of f at $\lambda = 1$. Furthermore, if rank $(I - Df(\bar{z})) = n$, then γ has finite arc length.

Proof: Since B is topologically equivalent to the closed unit ball, it suffices to consider $B = \{x \in \mathbb{R}^n \mid \|x\|_2 \leq 1\}$. Because of the rank assumption on $D_b a(b, z)$, $H(b, \lambda, z)$ is transversal to zero. Therefore, as in [15], by the Parameterized Sard's Theorem, $H_b(\lambda, z)$ is also transversal to zero for almost all $b \in U$. This fact, as shown in [15] or [16], implies the existence and nonintersection properties of γ. It also follows (again from [15] or [16]) that γ cannot just stop or wander around forever in $(0, 1) \times \operatorname{int} B$. Therefore γ must either return to $\lambda = 0$, penetrate the boundary of the cylinder $[0, 1) \times B$ at $0 < \lambda < 1$, or reach a point $(1, \bar{z})$. γ cannot return to $\lambda = 0$ because $a_b(z)$ had a unique nonsingular fixed point z_b. Consider any point $(\hat{\lambda}, \hat{z})$ on the boundary of $[0, 1) \times B$ where $0 < \hat{\lambda} < 1$, $\|\hat{z}\|_2 = 1$. Now (writing simply xy for the inner product of $x, y \in \mathbb{R}^n$)

$$\hat{z} H_b(\hat{\lambda}, \hat{z}) = (1 - \hat{\lambda})(\hat{z}\hat{z} - \hat{z} a_b(\hat{z})) + \hat{\lambda}(\hat{z}\hat{z} - \hat{z} f(\hat{z})) > 0$$

since $\hat{z}\hat{z} = 1$, $|\hat{z}a_b(\hat{z})| \leq \|\hat{z}\|_2\|a_b(\hat{z})\|_2 < 1$, and $|\hat{z}f(\hat{z})| \leq \|\hat{z}\|_2\|f(\hat{z})\|_2 \leq 1$. Therefore $H_b(\hat{\lambda}, \hat{z}) \neq 0$, and γ cannot penetrate the boundary for $0 < \hat{\lambda} < 1$. All that remains is that γ must reach a point $(1, \bar{z})$, at which $\bar{z} = f(\bar{z})$.

As in [15] or [16], the finite arc length of γ follows from the transversality of $H_b(\lambda, z)$ and the full rank of $I - Df(\bar{z})$. □

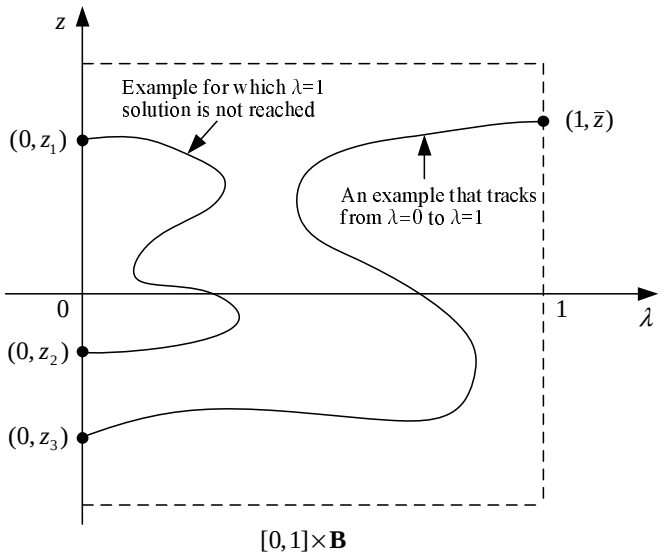


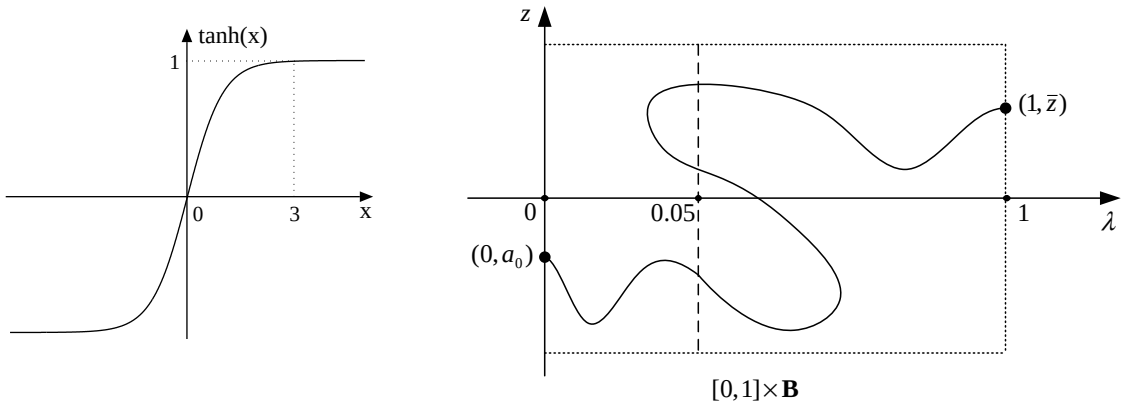
Fig. 1. Zero curve γ may return back to $\lambda = 0$ and there may not exist a path starting from some fixed points at $\lambda = 0$ and reaching a target solution at $\lambda = 1$.

The above theorem assumes that for each $b \in U$, $a_b(z)$ has a unique fixed point. If this condition does not hold, i.e., $a_b(z)$ has multiple fixed points, then as shown in the proof, it is not guaranteed that a fixed point $\bar{z}$ of f at $\lambda = 1$ can be reached. In this case, the zero curve γ may start from one fixed point of $a_b(z)$ and return back to another fixed point of $a_b(z)$ at $\lambda = 0$, as illustrated in Fig. 1. This is particularly relevant for the HMM, which generally is characterized by multiple $\lambda = 0$ (supervised) solutions, as discussed in Sec. IV.

To adapt to the case in which $a_b(z)$ has multiple fixed points, we therefore consider the third homotopy map by combining the homotopy maps in (2) and (3), to yield

$$H(\lambda, z) = (1 - \tanh 60\lambda)(z - a_0) + \tanh 60\lambda[(1 - \lambda)(z - a(z)) + \lambda(z - f(z))], \qquad (4)$$

where $a_0 \in \mathbb{R}^n$ is a constant vector, and $\tanh(\cdot)$ is the hyperbolic tangent function (see Fig. 2(a));



(a) (b)

Fig. 2. (a) The hyperbolic tangent function. (b) An example zero curve γ of the homotopy map (4): when $\lambda \in [0, 0.05)$, (4) behaves like (2); when $\lambda \in [0.05, 1)$, (4) is a close approximation to (3). Note that the range of $\lambda \in [0, 0.05)$ was intentionally enlarged to improve visibility.

the specific numbers in (4) may be altered slightly and still yield similar performance. The key point of (4) is that when $\lambda \in [0, 0.05)$, (4) behaves like the homotopy map (2) when regarding (4) as a convex combination with weight $\tau = \tanh(60\lambda)$, with the reachability of $\lambda = 0.05$ (equivalently, $\tau \approx 1$) guaranteed by Theorem 1; as λ moves beyond 0.05 and $\lambda \in [0.05, 1)$, (4) degenerates to the homotopy map (3), since $\tanh(60 \times 0.05) \approx 0.9951$. Technically, (4) is only an approximation to (3), but this approximation effectively converges as $\lambda \to 1$, since $\tanh(60{\times}0.3) = 1.0$ exactly with double precision arithmetic on any known computer. Given the analysis above, it can be demonstrated that the zero curve γ of (4) cannot return back to $\lambda = 0$, since it has a unique solution a_0 at $\lambda = 0$; however, γ can still visit $\lambda = 0.05$ multiple times, as shown in Fig. 2(b), until a fixed point $\bar{z}$ of f at $\lambda = 1$ is reached, with this guaranteed by Theorem 3. As discussed further below, these properties of (4) play a key role in semi-supervised HMM training, as well as in finding a preferential local optimum for the supervised learning.

Theorem 3: Let $B \subset \mathbb{R}^n$ be a compact, convex set with nonempty interior $\operatorname{int} B$, $U \subset \operatorname{int} B$ a nonempty open set, and let $f : B \to B$ and $a : B \to \operatorname{int} B$ be C^2 maps. Define $H : U \times [0, 1) \times B \to \mathbb{R}^n$ by

$$H(a_0, \lambda, z) = \left(1 - \tanh\left(\frac{60\lambda}{1-\lambda}\right)\right)(z - a_0) + \tanh\left(\frac{60\lambda}{1-\lambda}\right)\left[(1-\lambda)(z - a(z)) + \lambda(z - f(z))\right],$$

and define $H_{a_0}(\lambda, z) = H(a_0, \lambda, z)$. Then for almost all $a_0 \in U$ there is a zero curve γ of

$H_{a_0}(\lambda, z)$ emanating from $(0, a_0)$, along which the Jacobian matrix $DH_{a_0}(\lambda, z)$ has full rank, that does not intersect itself or other zeros of $H_{a_0}(\lambda, z)$, and reaches (accumulates at) a fixed point $\bar{z}$ of f at $\lambda = 1$. Furthermore, if rank $(I - Df(\bar{z})) = n$, then γ has finite arc length.

Proof: Since B is topologically equivalent to the closed unit ball, it suffices to consider $B = \{x \in \mathbb{R}^n \mid \|x\|_2 \leq 1\}$. Note than any $a_0 \in U \subset \operatorname{int} B$ satisfies $\|a_0\|_2 < 1$. It is trivially verified that $H(a_0, \lambda, z)$ is transversal to zero, and that rank $D_z H_{a_0}(0, z) = n$. Therefore, as in [15], by the Parameterized Sard's Theorem, $H_{a_0}(\lambda, z)$ is also transversal to zero for almost all $a_0 \in U$. This fact, as shown in [15] or [16], implies the existence and nonintersection properties of γ. It also follows (again from [15] or [16]) that γ cannot just stop or wander around forever in $(0, 1) \times \operatorname{int} B$. Therefore γ must either return to $\lambda = 0$, penetrate the boundary of the cylinder $[0, 1) \times B$ at $0 < \lambda < 1$, or reach a point $(1, \bar{z})$. γ cannot return to $\lambda = 0$ because $z = a_0$ is the unique nonsingular solution to $H_{a_0}(0, z) = 0$. Consider any point $(\hat{\lambda}, \hat{z})$ on the boundary of $[0, 1) \times B$ where $0 < \hat{\lambda} < 1$, $\|\hat{z}\|_2 = 1$. Now defining $\hat{\tau} = \tanh\big(60\hat{\lambda}/(1 - \hat{\lambda})\big)$ and writing simply xy for the inner product of $x, y \in \mathbb{R}^n$,

$$\hat{z} H_{a_0}(\hat{\lambda}, \hat{z}) = (1 - \hat{\tau})(\hat{z}\hat{z} - \hat{z}a_0) + \hat{\tau}\big[(1 - \hat{\lambda})(\hat{z}\hat{z} - \hat{z}a(\hat{z})) + \hat{\lambda}(\hat{z}\hat{z} - \hat{z}f(\hat{z}))\big] > 0$$

since $0 < \hat{\tau} < 1$, $\hat{z}\hat{z} = 1$, $|\hat{z}a(\hat{z})| \leq \|\hat{z}\|_2 \|a(\hat{z})\|_2 < 1$, $|\hat{z}a_0| \leq \|\hat{z}\|_2 \|a_0\|_2 < 1$, and $|\hat{z}f(\hat{z})| \leq \|\hat{z}\|_2 \|f(\hat{z})\|_2 \leq 1$. Therefore $H_{a_0}(\hat{\lambda}, \hat{z}) \neq 0$, and γ cannot penetrate the boundary for $0 < \hat{\lambda} < 1$. All that remains is that γ must reach a point $(1, \bar{z})$, at which $\bar{z} = f(\bar{z})$.

As in [15] or [16], the finite arc length of γ follows from the transversality of $H_{a_0}(\lambda, z)$ and the full rank of $I - Df(\bar{z})$. □

Observe that for $0.3 \leq \lambda \leq 1$, $\tanh(60\lambda) = \tanh\big(60\lambda/(1 - \lambda)\big) = 1.0$ exactly in 64-bit arithmetic. Therefore, as a practical matter, using $\tanh(60\lambda)$ is computationally equivalent to using $\tanh\big(60\lambda/(1 - \lambda)\big)$ in the homotopy map H_{a_0}.

Concerning the second issue with respect to the homotopy method, i.e., numerically tracking the zero path γ, there are three principal approaches for algorithmic implementations [16], with the typical time complexity of $\mathcal{O}(n^3)$, where n is the dimensionality of z. The three approaches are (i) solving an implicit ordinary differential equation, (ii) solving a rectangular system of equations (normal flow), and (iii) solving a square system of equations (augmented Jacobian

matrix). In all of these algorithms, the zero curve $\gamma = (\lambda(s), z(s))$ of H is parameterized by arc length s, and the algorithms essentially compute fixed points along γ by a predictor-corrector approach to solving $H(\lambda(s), z(s)) = 0$. All three algorithms can be found in HOMPACK90 [16], a suite of Fortran 90 codes for globally convergent homotopy algorithms, from which we choose the normal flow algorithm in the experiments that follow.

III. Semi-Supervised Generative Models and Mutual Information

The problem of interest here is the semi-supervised learning of a generative model (e.g., HMM). In this case both labeled and unlabeled data are assumed to be generated from an underlying joint density $p(\boldsymbol{x}, y|\Theta)$, where $\boldsymbol{x}$ is the feature vector, $y \in \{1, 2, \cdots, C, u\}$ is the corresponding class label and Θ is the model parameters. For sequential data, $\boldsymbol{x}$ represents the concatenation of a sequence of feature vectors. Note that a new category u augments the C classes, to denote a special class of data whose label is missing. Further, assume that the data are generated by the following two steps: (i) select a class according to the class prior probability $p(y|\Theta)$, and then (ii) generate a class-dependent feature vector $\boldsymbol{x}$ with distribution $p(\boldsymbol{x}|y, \Theta)$. For the case $y = u$, which means the label is missing (the corresponding feature vector is unlabeled), assume that the feature vector $\boldsymbol{x}$ is generated from the marginal density $p(\boldsymbol{x}|y = u, \Theta) = \sum_{y=1}^{C} p(\boldsymbol{x}, y|y \neq u, \Theta)$. Denote the probability of a data missing its label as $p(y = u|\Theta) = \lambda$. Then the likelihood of the model parameters Θ to have generated $(\boldsymbol{x}, y)$ can be expressed as

$$p(\boldsymbol{x}, y|\Theta) = [(1-\lambda)p(\boldsymbol{x}, y|y \neq u, \Theta)]^{\delta(y \neq u)} \times [\lambda p(\boldsymbol{x}|y = u, \Theta)]^{\delta(y=u)}, \tag{5}$$

where $\delta(\cdot)$ is the indicator function, with $\delta(e) = 1$ if e is true, and 0 otherwise.

The goal of semi-supervised learning is to estimate Θ from multiple samples $(\boldsymbol{x}, y)$, some of which may be unlabeled. This may be expressed in an information-theoretic setting in terms of the mutual information [18] as

$$MI(\boldsymbol{x}, y; \Theta) = En(\boldsymbol{x}, y) - En(\boldsymbol{x}, y|\Theta), \tag{6}$$

where En denotes the Shannon entropy, and Θ^* is sought to maximize the mutual information between samples of $(\boldsymbol{x}, y)$ and the model parameters Θ. This is equivalent to minimizing $En(\boldsymbol{x}, y|\Theta)$ or maximizing $E[\log p(\boldsymbol{x}, y|\Theta)]$, where in principle the expectation is over $\boldsymbol{x}$, y, and

Θ. In a maximum-likelihood (ML) setting for estimation of Θ, the probability density function of Θ is approximated by a point estimate, i.e., $p(\Theta) = \delta(\Theta = \Theta^*)$, with Θ^* representing the ML parameters. Then using (5) gives

$$\begin{aligned} E\left[\log p(\boldsymbol{x}, y|\Theta)\right] &= E\big\{\delta(y \neq u)\left[\log(1-\lambda) + \log p(\boldsymbol{x}, y|y \neq u, \Theta)\right] \\ &\qquad + \delta(y = u)[\log\lambda + \log p(\boldsymbol{x}|y = u, \Theta)]\big\} \\ &\propto (1-\lambda)E[\log p(\boldsymbol{x}, y|y \neq u, \Theta)] + \lambda E[\log p(\boldsymbol{x}|y = u, \Theta)], \end{aligned} \tag{7}$$

and Θ^* is sought to maximize

$$J_\lambda(\Theta) = (1-\lambda)E[\log p(\boldsymbol{x}, y|y \neq u, \Theta)] + \lambda E[\log p(\boldsymbol{x}|y = u, \Theta)]. \tag{8}$$

Given labeled data $\{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_L, y_L)\}$ and unlabeled data $\{\boldsymbol{x}_{L+1}, \cdots, \boldsymbol{x}_{L+U}\}$, this may be further approximated by

$$J_\lambda(\Theta) = (1-\lambda)\frac{1}{L}\sum_{i=1}^{L}\log p(\boldsymbol{x}_i, y_i|\Theta) + \lambda\frac{1}{U}\sum_{i=L+1}^{L+U}\log p(\boldsymbol{x}_i|\Theta), \tag{9}$$

where the expectations are replaced by the empirical estimates. Moreover, with $\lambda^* = U/(L+U)$, the empirical estimation of $p(y = u|\Theta)$, the objective function (9) reduces to the traditional optimality criterion for semi-supervised learning [3], [5], [7]:

$$J_{\lambda^*}(\Theta) = \sum_{i=1}^{L}\log p(\boldsymbol{x}_i, y_i|\Theta) + \sum_{i=L+1}^{L+U}\log p(\boldsymbol{x}_i|\Theta), \tag{10}$$

which maximizes the joint likelihood of both labeled data and unlabeled data.

However, without knowledge of the underlying true model, the selection of this value of λ is problematic, since for cases of limited labeled data (typically, $U \gg L$ and $\lambda \approx 1$) the algorithm almost becomes unsupervised, and therefore the labeled data (arguably the most useful data) are almost unused in a relative sense[2]. Therefore, a solution should be sought with another allocation $\lambda \in [0, 1)$ between a purely supervised learning ($\lambda = 0$) and a purely unsupervised learning ($\lambda = 1$), with the objective of identifying a proper λ when the data deviates from the model considered, even though this information is generally not *a priori* available.

[2]However, if the data truly comes from the model considered, maximum likelihood does the right thing, even if λ is so high. For instance, for a mixture of Gaussians, it is enough to have one labeled data per class to label the clusters, because unlabeled data alone can be used to estimate the clusters accurately.

A general approach to optimizing (9), for a given λ, is to use the EM algorithm [13]. In this procedure, the missing labels of the unlabeled data are treated as hidden variables, and the algorithm iteratively updates the model parameters Θ via the E step and the M step, until convergence to a local optimum. Typically, one may regard each EM iteration as a fixed point iteration of the form $\Theta^{t+1} = EM_\lambda(\Theta^t)$, which acts on the parameters of the current estimate Θ^t and produces another estimate Θ^{t+1} that monotonically increases the likelihood; a converged solution of the EM algorithm is a fixed point of an EM operator. Since the homotopy method can be used to find the fixed points of a nonlinear system of equations, this point of view motivates applying the homotopy method to the EM operator.

IV. Semi-Supervised Learning of HMMs via a Homotopy Method

The detailed fixed point EM operator for solving (9) depends on the model used. For the problem of interest here, we consider the case of HMMs for semi-supervised training, and use the homotopy method to track the fixed point solutions from $\lambda = 0$ to $\lambda = 1$.

A. HMM classifier

Using similar notation as in [8], we define an N-state discrete HMM with an observation alphabet size of M, parameterized as $\theta = \{\pi^N, A^{N\times N}, B^{N\times M}\}$, where π is the initial-state probability vector, A is the state-transition matrix, and B is the observation matrix (probability of observing each of the M alphabet members in a particular state). Given an observation sequence $\boldsymbol{x} = \{x_1, \cdots, x_T\}$, the likelihood of the model θ is calculated as

$$p(\boldsymbol{x}|\theta) = \sum_{\boldsymbol{s}} \pi_{s_1} \cdot \prod_{t=1}^{T-1} a_{s_t s_{t+1}} \cdot \prod_{t=1}^{T} b_{s_t}(x_t), \tag{11}$$

where $\boldsymbol{s} = (s_1, \cdots, s_T)$ is the hidden state sequence, and the summation is over all possible state sequences.

For a sequential data classification task, an HMM classifier is built to assign a class label for a given data sequence. Based on the generative assumption in Sec. III, in this case each class will, in general, be modeled as an HMM with a distinct set of parameters, i.e., $\theta^y = \{\pi^y, A^y, B^y\}$, for

each $y \in \{1, 2, \cdots, C\}$. By incorporating the class prior $\{w^1, \cdots, w^C\}$, the posterior distribution of class label y can be computed via Bayes rule as

$$p(y|\boldsymbol{x}, \Theta) = \frac{w^y p(\boldsymbol{x}|\theta^y)}{\sum_{y'=1}^{C} w^{y'} p(\boldsymbol{x}|\theta^{y'})} \tag{12}$$

where $0 \leq w^y \leq 1$, for all $y \in \{1, 2, \cdots, C\}$, and $\sum_{y=1}^{C} w^y = 1$, and $\Theta = \{w^1, \theta^1, \cdots, w^C, \theta^C\}$ denotes the cumulative parameters of the HMM classifier. The class that has the highest posterior probability is used as the estimated class label.

B. Semi-supervised HMM training via a homotopy method

Given the labeled data sequences $\{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_L, y_L)\}$ and unlabeled data sequences $\{\boldsymbol{x}_{L+1}, \cdots, \boldsymbol{x}_{L+U}\}$, the parameters of the HMM classifier Θ can be estimated by maximizing the objective function (9), which may be implemented by the homotopy method on the fixed point EM operator, as discussed in Sec. III.

Applying the EM algorithm to semi-supervised HMM classifier training, i.e., to objective function (9), is straightforward. The resulting algorithm is a simple extension of the standard Baum-Welch algorithm to handle the unlabeled data (e.g., see [7]). This yields the following fixed point equations:

$$\tilde{w}^y = (1-\lambda)\frac{1}{L}\sum_{i=1}^{L} \delta(y=y_i) + \lambda\frac{1}{U}\sum_{j=L+1}^{L+U} p(y|\boldsymbol{x}_j, \Theta), \tag{13}$$

$$\tilde{\pi}_k^y = (1-\lambda)\frac{1}{L}\sum_{i=1}^{L} \gamma_1^i(k)\delta(y=y_i) + \lambda\frac{1}{U}\sum_{j=L+1}^{L+U} \gamma_1^j(k) p(y|\boldsymbol{x}_j, \Theta), \tag{14}$$

$$\begin{aligned}\tilde{a}_{kl}^y = (1-\lambda)\frac{1}{L}\sum_{i=1}^{L}\sum_{t=1}^{T-1} \xi_t^i(k,l)\delta(y=y_i) \\ + \lambda\frac{1}{U}\sum_{j=L+1}^{L+U}\sum_{t=1}^{T-1} \xi_t^j(k,l) p(y|\boldsymbol{x}_j, \Theta),\end{aligned} \tag{15}$$

$$\begin{aligned}\tilde{b}_k^y(v) = (1-\lambda)\frac{1}{L}\sum_{i=1}^{L}\sum_{t=1}^{T} \gamma_t^i(k)\delta(x_{i,t}=v)\delta(y=y_i) \\ + \lambda\frac{1}{U}\sum_{j=L+1}^{L+U}\sum_{t=1}^{T} \gamma_t^j(k)\delta(x_{j,t}=v) p(y|\boldsymbol{x}_j, \Theta),\end{aligned} \tag{16}$$

where the left hand sides of the above equations are unnormalized parameters of an HMM classifier, related to Θ by

$$\pi_k^y = \frac{\tilde{\pi}_k^y}{\sum_i \tilde{\pi}_i^y} \tag{17}$$

with $\tilde{\pi}_i^y \geq 0$, for all $i \in \{1, \cdots, N\}$, $y \in \{1, \cdots, C\}$, and similarly for the parameters w^y, a_{kl}^y and $b_k^y(v)$. Note that with $\lambda = 0$, the fixed point equations in (13)–(16) after normalization degenerate to the standard Baum-Welch algorithm [8] for supervised HMM parameter updating.

Summarizing the HMM fixed point equations (13)–(16) in matrix form, we have

$$H(\lambda, \tilde{\Theta}) = (1 - \lambda)(\tilde{\Theta} - EM_0(\Theta)) + \lambda(\tilde{\Theta} - EM_1(\Theta)) = 0, \tag{18}$$

where $\tilde{\Theta} = \{\tilde{w}^1, \tilde{\theta}^1, \cdots, \tilde{w}^C, \tilde{\theta}^C\}$ is an unnormalized version of Θ, $EM_0(\Theta)$ are the right hand side terms of (13)–(16) when $\lambda = 0$, and $EM_1(\Theta)$ are the right hand side terms of (13)–(16) when $\lambda = 1$. Note that (18) is in the same form as (3), forming a "natural-parameter homotopy" (where here $\tilde{\Theta}$ represents the variable z used when introducing the homotopy method, and b may be regarded as choosing labeled data for the computation of $EM_0(\Theta)$).

We emphasize that the EM algorithm is employed for supervised HMM training ($\lambda = 0$) with the state sequences treated as the hidden variables. For semi-supervised HMM training, the EM algorithm becomes more involved, since in this case there are two levels of hidden variables: one is the hidden state sequences for the labeled and unlabeled data, and the other is the hidden labels of the unlabeled data. Because of these hidden variables, both the supervised solution ($\lambda = 0$) and the unsupervised solution ($\lambda = 1$) are characterized by multiple local optima. As discussed in Sec. II, in this case if we use the homotopy map (3) for path tracking, it is not guaranteed that a fixed point solution at $\lambda = 1$ can be reached, and the zero curve γ may start from one fixed point of $EM_0(\Theta)$ and return back to another fixed point of $EM_0(\Theta)$ at $\lambda = 0$ (see Fig. 1). For this reason, when we implement the homotopy method for semi-supervised HMM training, we use the homotopy map (4), which is a close approximation to (3) but without problems posed by multiple local-optimal supervised solutions, and a smooth nonbifurcating path of fixed points solution is assured, with probability one, from $\lambda = 0$ to $\lambda = 1$.

To this end, we transform (18) into the form of (4), and have

$$\begin{aligned} H(\lambda, \tilde{\Theta}) &= (1 - \tanh 60\lambda)(\tilde{\Theta} - \Theta_0) \\ &\quad + \tanh 60\lambda \left[(1-\lambda)(\tilde{\Theta} - EM_0(\Theta)) + \lambda(\tilde{\Theta} - EM_1(\Theta)) \right], \end{aligned} \tag{19}$$

where Θ_0 is a constant vector, which we here set to an arbitrarily chosen fixed point of $EM_0(\Theta)$, corresponding to selecting a local supervised solution. To implement the homotopy algorithm, the Jacobian matrices $\nabla_{\tilde{\Theta}} EM_0(\Theta)$ and $\nabla_{\tilde{\Theta}} EM_1(\Theta)$ are required, which subsequently require the partial derivatives of $\gamma_t(i)$, $\xi_t(i,j)$ and $p(y|\boldsymbol{x}, \Theta)$ with respect to each element of $\tilde{\Theta}$. However, the currently available variables in HMMs, such as $\gamma_t(i)$ and $\xi_t(i,j)$, are not amenable to the computation of these partial derivatives. We thus define a set of new variables ψ, ϕ, and Φ, as presented in Appendix I, to facilitate this computation, and the final computational formulas for $\nabla_{\tilde{\Theta}} EM_0(\Theta)$ and $\nabla_{\tilde{\Theta}} EM_1(\Theta)$ are summarized in Appendix II.

We note that for the relatively simple graphical models, such as naive Bayes and Gaussian mixture models (GMM), as used by Corduneanu and Jaakkola in [14], the supervised solution ($\lambda = 0$) is unique. Therefore, the associated fixed point equations can be formulated in the form of (2), with a strong theoretical existence guarantees provided by Theorem 1. From this point of view, our application of the homotopy map (4) extends the previous work of Corduneanu and Jaakkola to a more general case of semi-supervised learning for generative models.

C. Determination of parameter λ

As discussed in Sec. II, the homotopy map (4) is a close approximation to (3), without problem posed by the multiple local optima of the supervised HMM, and assuring with probability one a track from $\lambda = 0$ to $\lambda = 1$. While we have found this successful track to be the case for the HMM, interesting numerical phenomena have been observed, as demonstrated in Sec. V. Specifically, for many $\lambda = 0$ initializations, each representing a particular supervised (local optimal) HMM solution, we observe a homotopy zero curve tracking away from $\lambda = 0$ and then backtracking to the vicinity of another $\lambda = 0$ solution, which we attribute to a different local optimal supervised solution (see Fig. 3(a)). The λ then increases away from this local optimal supervised solution (at $\lambda = 0$), often followed by tracking back to near a different $\lambda = 0$

solution. This "hopping" between different supervised solution neighborhoods (at $\lambda = 0$) often occurs many times, before there is a final departure of the homotopy zero curve track from the last $\lambda = 0$ solution neighborhood, followed by a smooth track to a $\lambda = 1$ (unsupervised) solution.

This phenomenon may be interpreted as follows. The supervised HMM analysis permits multiple local-optimal solutions. For cases of limited labeled data, it is anticipated that there may be more such local optimal solutions, with enhanced uncertainty as to which is appropriate. When λ moves away from $\lambda = 0$ in the homotopy zero curve, the algorithm begins to sense the properties of the (abundant) unlabeled data. If the $\lambda = 0$ solution corresponds to a supervised solution that is inconsistent with the unlabeled data, the homotopy zero curve tracks back to near a different $\lambda = 0$ solution that is better matched to the unlabeled data. This tracking away from $\lambda = 0$ solution neighborhoods and back to different $\lambda = 0$ solution neighborhoods may occur multiple times before an appropriate $\lambda = 0$ solution neighborhood is found, after which the homotopy method tracks smoothly to the unsupervised $\lambda = 1$ solution.

This interpretation, which we support with data in Sec. V, indicates that the homotopy method is selecting a supervised ($\lambda = 0$) solution (by visiting its neighborhood) that is well matched to the unlabeled data. As discussed in Sec. V, the parameters associated with this $\lambda = 0$ solution may therefore be desirable parameters for the HMM classifiers of interest. In addition, as the main goal of this paper, we wish to choose a proper $\lambda \in [0, 1)$ for semi-supervised learning, rather than simply selecting from the multiple $\lambda = 0$ solutions. In this case we select λ based on an analysis of the final complete track to $\lambda = 1$, from the selected $\lambda = 0$ solution, since based on the above discussion this track constitutes the case of an appropriate match between the supervised and unsupervised solutions. We refer to this final zero curve segment from the vicinity of the last $\lambda = 0$ to $\lambda = 1$ as P_F.

For each λ along P_F, we have a fixed point solution for the parameters of the HMM classifier Θ_λ. Therefore, for each λ we may compute the average classification uncertainty for the unlabeled data, quantified via the entropy

$$En(Y|\Theta_\lambda) = -\frac{1}{U} \sum_{i=L+1}^{L+U} \sum_{y=1}^{C} p(y|\boldsymbol{x}_i, \Theta_\lambda) \log p(y|\boldsymbol{x}_i, \Theta_\lambda). \tag{20}$$

As demonstrated in Sec. V, we have found that the λ that maximizes (20) along P_F provides a good estimate for the parameters of the HMM classifier. The λ that maximizes (20) implies a point of greatest classification uncertainty, which may be counterintuitive, as it conflicts with the general belief of minimizing classification uncertainty. The λ that maximizes (20) constitutes a point at which the balance between labeled and unlabeled data introduces the least classification confidence. This is analogous to the analysis performed in [14], in which the λ operating point was determined as the point at which there was an apparent transition between the purely supervised and purely unsupervised solutions. As demonstrated in Sec. V this selection procedure for λ has yielded good performance on the data considered thus far, but further investigation of this measure is warranted.

We also note that the λ selected through maximization of (20) yielded comparable performance to a minimax solution applied to the functional in (9), where in that case we maximize with respect to HMM parameters Θ and minimize with respect to the allocation parameter λ. For this reason, maximization of (20) may be related to the robustness of the minimax estimate [19], an important technique for designing systems that are robust with respect to modeling uncertainties. Because of the strong connection between minimax optimization and game theory [19], we also suggest the following game-theoretic view of semi-supervised learning. From this view, both labeled data and unlabeled data may take competitive roles to optimize the objective function (9). Such a game-theoretic perspective is not considered further here, but may be an interesting direction for future work.

V. Experimental Results

We test the performance of the homotopy method for HMM classifier design on measured acoustic data and synthetic data. As discussed above, the homotopy method not only allows one to select a proper $\lambda \in [0, 1)$ for semi-supervised learning, it also provides a means of selecting a local supervised solution that appears to be consistent with the unlabeled data; both supervised solution and semi-supervised solution are sought on the final zero curve segment P_F.

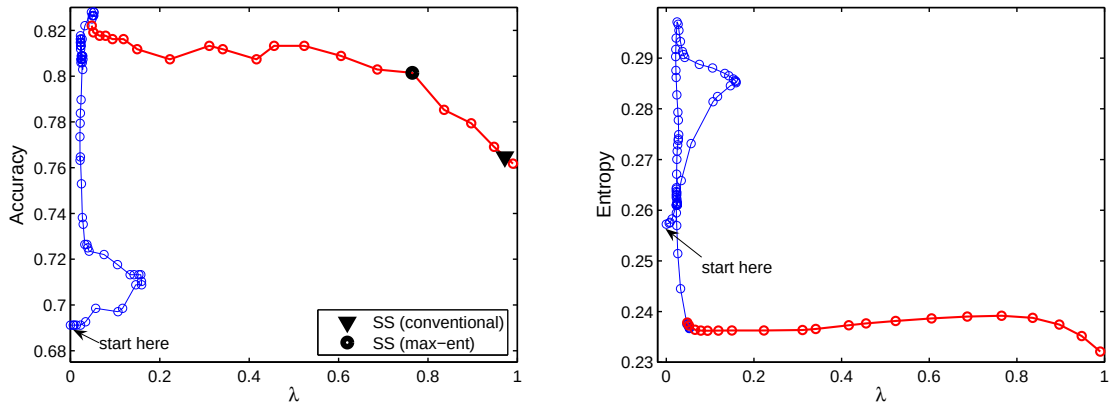
Based on previous studies, it is anticipated that the conventional semi-supervised solution will work well for synthetic data, for which the model fits perfectly [6]. However, for the HMM

modeling of measured data, for which the model deviation does exist, the performance of the conventional semi-supervised solution is expected to degrade, and it is here that the homotopy method is expected to excel. Both the measured data and synthetic data used in the experiments are available at `http://www.ee.duke.edu/~lcarin/homotopy_HMMs.zip`.

A. Experiments on measured data

We consider a multi-aspect target classification task based on measured acoustic scattering data. Details on using HMMs for this application may be found in [9]. We here provide the basic idea of multi-aspect classification and a brief description of the characteristics of the measured data. Typically, the acoustic fields scattered from a complex target are a strong function of target-sensor orientation. However, there are often sets of contiguous target-sensor orientations for which the scattering data are relatively stationary, with each such set termed a target "state". When sensing is performed from a sequence of target-sensor orientations, one implicitly senses scattered fields sampled from a sequence of target states, and this sequence of sampled states may be modeled as a Markov process [9]. Since the target is generally distant or concealed, the underlying sampled states are unobserved, or "hidden", and only the associated scattered fields are observed. This therefore yields the aforementioned HMM representation of the scattering data. For each target-sensor orientation the associated acoustic scattered fields are mapped to a feature vector, and then this feature vector is quantized using vector quantization (VQ) [20]. The sequence of measured scattered waveforms is therefore mapped to a sequence of code indices, modeled via a discrete HMM. Details on the targets and on the feature extraction employed may be again found in [9].

In the experiments each target is modeled as a 2-state HMM with an observation alphabet size of 10. The task concerns classifying between two targets based on a sequence of 8 observations (corresponding in the physical problem to viewing the target from eight orientations, at 5° angular sampling, for a total aperture of 35°). The original data are sampled at 1° angular increment, and therefore there are a total of 360 data sequences for each target, defined by the initial angle of orientation. We randomly select 10 data sequences as the labeled data, and use the remaining 350 data sequences as the unlabeled data. Therefore, there are totally $L = 20$ labeled data



(a) (b)

Fig. 3. An example path tracking by using the homotopy map (4), with the initial part of the path shown in blue, and the final track P_F shown in red. (a) Classification accuracy on the unlabeled data as a function of λ; (b) classification entropy on the unlabeled data as a function of λ. For semi-supervised learning, we consider selection of λ by the "conventional" method $\lambda = U/(L+U)$ and the maximum entropy criterion in (20).

sequences (10 for each of the two targets) and $U = 700$ unlabeled data sequences, on which a semi-supervised HMM classifier is trained.

We use the normal flow algorithm, as implemented in the subroutine STEPNF from HOMPACK90 [16], to track the fixed point solutions of (19) for various allocations $\lambda \in [0, 1)$. Each λ along the track is determined by the algorithm that starts at one randomly selected fixed point solution of $EM_0(\Theta)$ and tracks the fixed point solutions along the zero curve γ. For each obtained fixed point solution (at each reachable λ), we obtain the parameters of the HMM classifier Θ_λ, via the normalization as in (17). This classifier is then applied on the unlabeled data to evaluate classification accuracy.

Figure 3 shows an example result of the homotopy method for path tracking of the semi-supervised HMM. The evolution of classification accuracies as a function of λ for the homotopy map (4) is shown in Fig. 3(a), and the entropy from (20) is shown in Fig. 3(b). In the plot the initial portion of the homotopy zero curve is shown in blue, and the final zero curve segment P_F from the vicinity of the last $\lambda = 0$ to $\lambda = 1$ is shown in red. It is demonstrated in Fig. 3(a) that the homotopy method initially tracks between different solutions near $\lambda = 0$, before completing a path to $\lambda = 1$, and the classification performance for the $\lambda \approx 0$ solutions, as measured on the

unlabeled data, increases with successive visits to $\lambda \approx 0$ solutions. The last visited $\lambda \approx 0$ solution, at the start of P_F, may be a good operating point for the HMMs, corresponding to selection of a preferential (local) supervised solution that appears to be consistent with the unlabeled data. We do indeed consider supervised HMMs, with EM-determined parameters initialized at values given at the start of P_F, and this yields very similar classification accuracy as the last $\lambda \approx 0$ solution, indicating the homotopy determined $\lambda \approx 0$ solution is indeed at the vicinity of a supervised solution. Although this final HMM solution is purely supervised, the unlabeled data have been used to determine a good EM initialization point, thereby selecting from among the multiple $\lambda = 0$ solutions. It is important to emphasize that in practice the classification performance information would be unavailable and cannot be used to select λ for semi-supervised learning. We therefore consider the semi-supervised learning by selecting λ based on the "conventional" method $\lambda = U/(L+U)$ and the maximum entropy criterion in (20), with the selection is only considered on the final zero curve segment P_F. From Fig. 3 we find that the maximum entropy criterion chooses $\lambda \approx 0.75$, while the "conventional" method chooses $\lambda \approx 0.97$.

For a comprehensive demonstration of the homotopy-based semi-supervised HMM, in Fig. 4 we present three additional set of example results that are representative from 60 random runs. We use the same dataset and experimental setting as in Fig. 3, except that the labeled data in each run are randomly selected. The top and bottom examples in Fig. 4 are similar to Fig. 3, featured by backtracking to another local supervised solution before the final direct track to $\lambda = 1$. While the top example is the most general case, the bottom example does exist but happens infrequently. Of the 60 random runs, we also note some cases like in the middle example, for which the initial $\lambda = 0$ solution yields a monotone increasing (in λ) track to $\lambda = 1$. For cases like the middle example, sometimes the $\lambda = 0$ solution yields the highest classification accuracy, while other times (like in Fig. 4(c)) does not. However, in all the cases the maximum entropy criterion in (20) yields a good estimate of λ for semi-supervised learning.

To provide a statistically meaningful analysis, in Table I we present the average performance from the aforementioned 60 random runs, with the average performance relative to the initial ($\lambda = 0$) supervised solution. The results of three different algorithms are presented. In the first we consider the last visited $\lambda \approx 0$ solution, at the start of P_F. This solution is supervised, but the

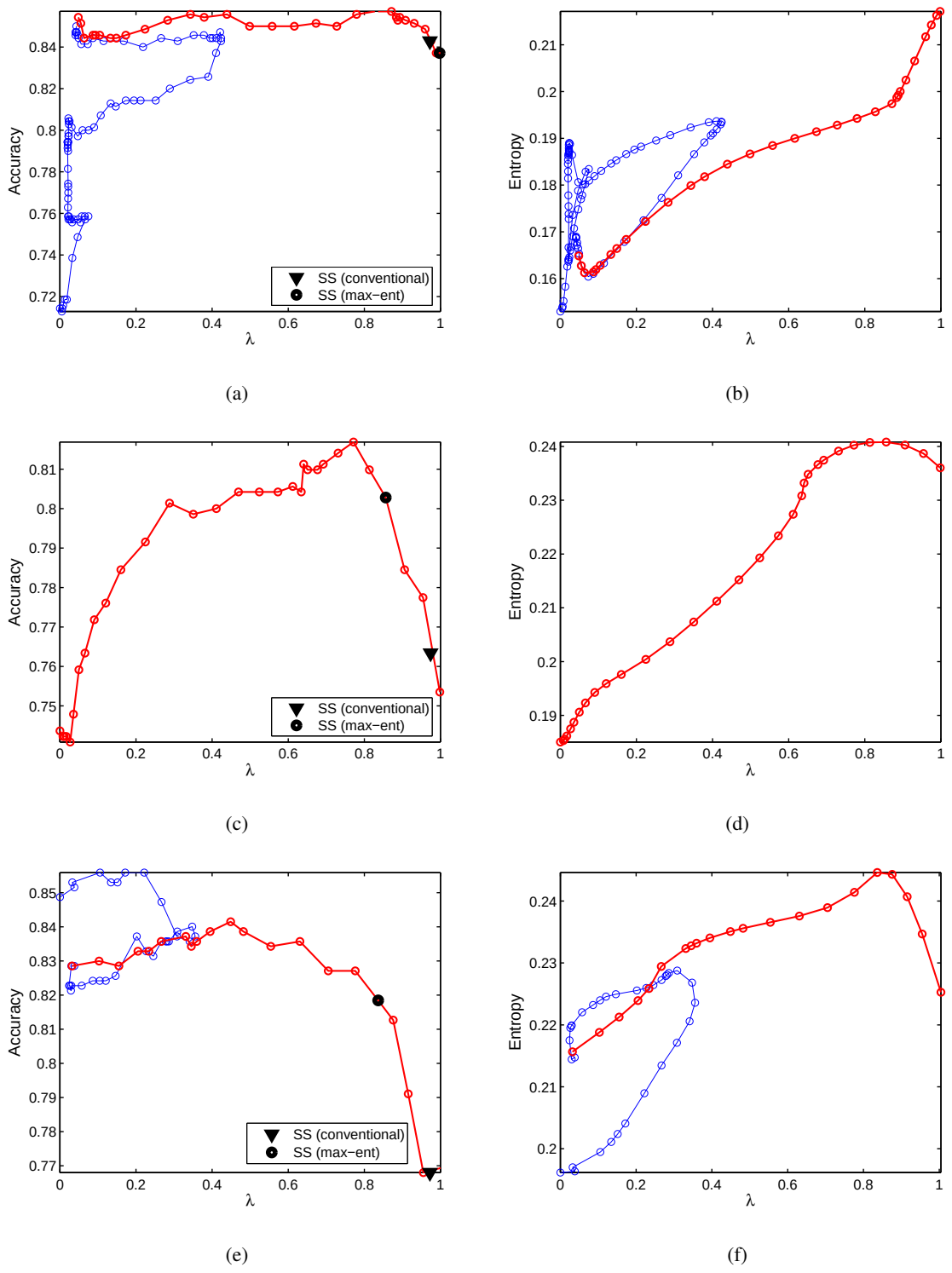


Fig. 4. Three sets of representative results from 60 random runs. The three cases are shown vertically, with the left figure showing the classification accuracy as a function of λ, and the right showing the entropy in (20) as a function of λ. We indicate in (a), (c) and (e) the λ selection criteria based on "conventional" $\lambda = U/(L+U)$, as well as the maximum entropy in (20).

TABLE I

AVERAGE CLASSIFICATION PERFORMANCE RELATIVE TO THE INITIAL $\lambda = 0$ SOLUTION. RESULTS ARE AVERAGED OVER A TOTAL OF 60 EXAMPLES, FOR WHICH 10 OF 360 DATA SEQUENCES ARE SELECTED RANDOMLY, AND TREATED AS LABELED DATA, THE REMAINING 350 TREATED AS UNLABELED. FOR THE HOMOTOPY COMPUTATIONS, OF THE 60 CASES 36 EXPERIENCED BACKTRACKING TO A $\lambda \approx 0$ SOLUTION, AS IN FIG. 3. AVERAGE RELATIVE CLASSIFICATION IMPROVEMENT IS SHOWN FOR THE CASES IN WHICH HOMOTOPY MANIFESTS AT LEAST ONE RETURN TO A $\lambda \approx 0$ SOLUTION, WHEN IT DOES NOT, AND AVERAGE OVERALL PERFORMANCE.

	last $\lambda \approx 0$	$\lambda = U/(L+U)$	λ_{ME}
with return to $\lambda \approx 0$	4.95%	3.10%	6.19%
without return to $\lambda \approx 0$	0	-2.86%	0.98%
Total	3.02%	0.72%	4.11%

unlabeled data are used via homotopy to select a preferential local ($\lambda \approx 0$) supervised solution. In the second we consider the conventional semi-supervised approach using $\lambda = U/(L+U)$; this is the "conventional" solution shown in Figs. 3 and 4. Finally, we consider a semi-supervised solution at λ_{ME}, computed by maximizing (20); this is the "max-ent" solution in Figs. 3 and 4. All three algorithms are again evaluated along the final zero curve segment P_F.

Of the 60 random runs considered, we observed backtracking of the type in Fig. 3 on 36 cases, where in 24 cases no backtracking of the homotopy solution was observed, as in Fig. 4(c). In Table I we present results when the homotopy does have a return to $\lambda \approx 0$ (backtracking), and when it does not, as well as total average performance. It is observed that, for these data, the λ_{ME} semi-supervised solution provides the largest improvement relative to the initial supervised solution.

For an extensive performance comparison between supervised learning and semi-supervised learning, in Fig. 5 we compare the performance of both algorithms on an increasing number of labeled data, using the same data set as considered above. The supervised learning uses the EM algorithm, whose solution is also used as the initial $\lambda = 0$ solution for the homotopy tracking; and the semi-supervised learning use the homotopy method, with the λ determined either by the "conventional" method $\lambda = U/(L+U)$ or by the maximum entropy criterion in (20). We implement the experiments 10 times, with labeled data randomly selected in each time, and the

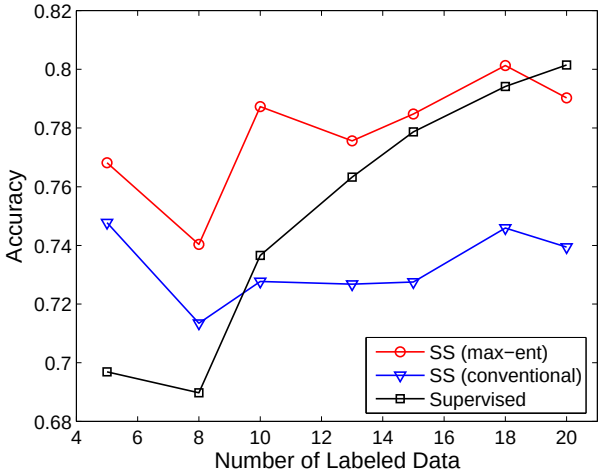


Fig. 5. Classification accuracy on measured data as a function of labeled sequences per target, for the supervised learning, the conventional semi-supervised learning with $\lambda = U/(L+U)$, and the homotopy method based on the maximum entropy in (20).

average performance is reported in Fig. 5. It is observed that the maximum entropy criterion in (20) yields superior performance compared to $\lambda = U/(L+U)$ for these measured data, for which the model is not perfect, and the λ_{ME} semi-supervised learning outperforms the supervised learning when the amount of labeled data is scarce, and the performances of two get close as the number of labeled data increases.

B. Experiments on synthetic data that match the model

In the last set of experiments we test the performance of the homotopy method on a synthetic data set. The data set considered includes 2 classes, each of which has 500 data sequences of length 5 that are generated from a 2-state HMM with an observation alphabet size of 5. We use the same experimental setting as in Fig. 5, with the average performance over 10 random runs shown in Fig. 6. In this case the data were generated from the same model used for analysis, and good performance is observed when using $\lambda = U/(L+U)$.

It is demonstrated in Fig. 6 that for the case in which the data fits the model, the semi-supervised solution with $\lambda = U/(L+U)$ yields the best performance, while the maximum entropy criterion in (20) yields the results that are slightly worse. However, both semi-supervised solutions are

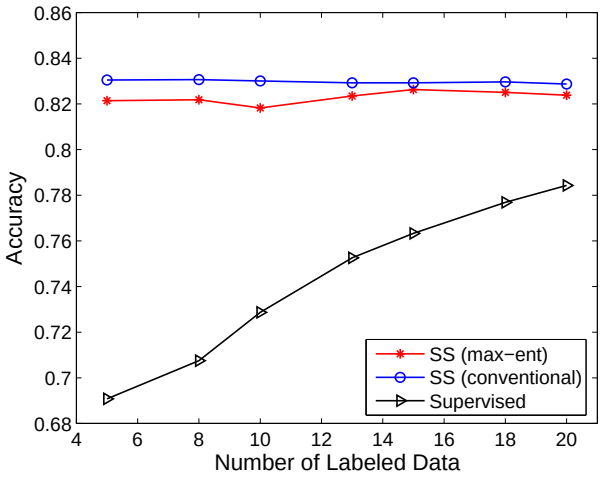


Fig. 6. Classification accuracy on synthetic data as a function of labeled sequences per class, for the supervised learning, the conventional semi-supervised learning with $\lambda = U/(L+U)$, and the homotopy method based on the maximum entropy in (20).

significantly better than the supervised learning, particularly for small L.

As a last note, we emphasize that when using $\lambda = U/(L+U)$ the homotopy method *may* provide a different solution than previous work in which λ was set this way. Specifically, previously research [7] solved (13)–(16) with $\lambda = U/(L+U)$ directly. By contrast, in the homotopy method we track solutions from $\lambda = 0$ to $\lambda = 1$, and select one fixed point solution at $\lambda = U/(L+U)$. Consequently, even when using $\lambda = U/(L+U)$ there is a distinction between the homotopy formulation developed here and previous semi-supervised solutions for HMMs. For the case for which the model matches the data, as in Fig. 6, we have found using $\lambda = U/(L+U)$ in (13)–(16) and within the homotopy formulation yield similar results, while this is not the case when the model does not match perfectly (as in Fig. 5).

VI. Conclusion

In this paper we have considered semi-supervised learning of HMMs, based on a globally convergent probability-one homotopy method that yields a path of fixed point HMM solutions, each utilizing a different balance of labeled and unlabeled data, dictated by a parameter λ. A significant challenge involved addressing the multiple local optimal solutions afforded by the

supervised HMM solution. To address this problem a new probability-one homotopy map was constructed, which allowed analysis of the desired semi-supervised problem, without problems posed by multiple local-optimal supervised solutions.

This homotopy formulation yielded interesting phenomena based on analysis of measured sequential data (for which the model is not perfect). We observed that the homotopy algorithm often tracked away from one supervised ($\lambda = 0$) solution neighborhood to another one, until manifesting a final track from a last supervised solution neighborhood to the purely unsupervised ($\lambda = 1$) solution. We referred to this as the "final" homotopy path segment P_F. Based on a detailed analysis of performance, we attributed this phenomena to a homotopy-based examination of multiple supervised solutions, until a solution is found that is in agreement with the properties of the unlabeled data, and from this supervised solution neighborhood there is a final track to the unsupervised solution at $\lambda = 1$.

Having developed the homotopy method, there is now the issue of choosing from among the numerous fixed point solutions along the homotopy path, with this solution used in the final classification. Three different ways were considered for choosing this single fixed point solution: (i) $\lambda = U/(L + U)$, where L and U represent the number of labeled and unlabeled data, respectively; (ii) using the $\lambda \approx 0$ solution at the beginning of the final homotopy path segment P_F; and (iii) considering the λ along P_F that maximizes the classification uncertainty, computed in terms of the entropy. We found that methods (ii) and (iii) yielded superior results compared to (i), when considering data for which there was not a perfect match between the data and model. Methods (ii) and (iii) yielded similar results, although on average (iii) was slightly better.

For models like the HMM, for which there are multiple local supervised solutions, further research is required on selecting the optimal λ. Specifically, at this point we use the maximum entropy criterion to select λ, which yielded performance comparable to that of a related minimax analysis [19]. A more effective perspective, inspired by minimax optimization, to semi-supervised learning may be game theoretic, in which both labeled data and unlabeled data take competitive roles to optimize the objective function (9). A more detailed theoretical analysis in this framework is warranted for future research.

REFERENCES

[1] S. J. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 252–264, Mar. 1991.

[2] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. of the 20th International Conference on Machine Learning*, no. 912-919, 2003.

[3] M. Seeger, "Learning with labeled and unlabeled data," Institute for Adaptive and Neural Computation, University of Edinburgh, Edinburgh, UK, Tech. Rep., 2001.

[4] B. Shahshahani and D. Landgrebe, "The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon," *IEEE Trans. on Geosc. and Remote Sensing*, vol. 32, 1994.

[5] K. Nigam, A. Mccallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Machine learning*, vol. 39, pp. 135–167, 2000.

[6] F. G. Cozman, I. Cohen, and M. C. Cirelo, "Semi-supervised learning of mixture models," in *Proc. of the 20th International Conference on Machine Learning*, 2003.

[7] M. Inoue and N. Ueda, "Exploitation of unlabeled sequences in hidden Markov models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, 2003.

[8] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[9] P. Runkle, P. K. Bharadwaj, L. Couchman, and L. Carin, "Hidden Markov models for multi-aspect target classification," *IEEE Trans. Signal Proc.*, vol. 47, pp. 2035–2040, Jul. 1999.

[10] E. Birney, "Hidden Markov models in biological sequence analysis," *IBM Journal of Research and Development*, vol. 45, 2001.

[11] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun, "A practical part-of-speech tagger," in *Proc. Third Conf. Applied Natural Language Processing*, 1992, pp. 133–140.

[12] D. M. Bikel, R. Schwartz, and R. M. Weischedel, "An algorithm that learns what's in a name," *Machine Learning*, vol. 34, pp. 211–231, Feb. 1999.

[13] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[14] A. Corduneanu and T. Jaakkola, "Continuation methods for mixing heterogeneous sources," in *Proc. of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2002.

[15] S. N. Chow, J. Mallet-Paret, and J. A. Yorke, "Finding zeros of maps: Homotopy methods that are constructive with probability one," *Math. Comput.*, vol. 32, pp. 887–899, 1978.

[16] L. T. Watson, M. Sosonkina, R. C. Melville, A. P. Morgan, and H. F. Walker, "Algorithm 777: HOMPACK90: A suite of Fortran 90 codes for globally convergent homotopy algorithms," *ACM Trans. Math. Software*, vol. 23, pp. 514–549, 1997.

[17] E. L. Allgower and K. Georg, *Numerical continuation methods: An introduction*. New York: Springer-Verlag, 1990.

[18] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY: Wiley, 1991.

[19] S. Verdú and H. V. Poor, "On minimax robustness: A general approach and applications," *IEEE Trans. Information Theory*, vol. 30, no. 2, pp. 328–340, Mar. 1984.

[20] R. M. Gray, "Vector quantization," *IEEE ASSP Magazine*, pp. 4–29, Apr. 1984.

APPENDIX I
NEW HMM VARIABLES

The basic variables in HMMs, such as $\gamma_t(i) = p(s_t{=}i|o_1, \ldots, o_T)$ and $\xi_t(i,j) = p(s_t{=}i, s_{t+1}{=}j|o_1, \ldots, o_T)$, are not amenable to computing the Jacobian matrices for the implementation of the homotopy method. We therefore define a set of new HMM variables ψ, ϕ, and Φ to facilitate this computation.

For $t = 1, \cdots, T$ and $t' = 1, \cdots, t{-}1$, we define

$$\psi^L_{t,t'}(i,j) = p(o_{t'+1}, \ldots, o_{t-1}, s_t = i|s_{t'} = j), \tag{21}$$

$$\phi^L_{t,t'}(i,j) = p(s_t = i|o_{t'+1}, \ldots, o_T, s_{t'} = j), \tag{22}$$

and for $t = 1, \cdots, T{-}1$ and $t' = 1, \cdots, t$,

$$\Phi^L_{t,t'}(i,j,k) = p(s_t = i, s_{t+1} = j|o_{t'+1}, \ldots, o_T, s_{t'} = k), \tag{23}$$

where the superscript L indicates that $t' \leq t$.

Similarly, for $t = 1, \cdots, T$ and $t' = t{+}1, \cdots, T$, we define

$$\psi^R_{t,t'}(i,j) = p(o_{t+1}, \ldots, o_{t'-1}, s_{t'} = j|s_t = i) = \psi^L_{t',t}(j,i), \tag{24}$$

$$\phi^R_{t,t'}(i,j) = p(s_t = i|o_1, \ldots, o_{t'-1}, s_{t'} = j), \tag{25}$$

and for $t = 1, \cdots, T{-}1$ and $t' = t{+}1, \cdots, T$,

$$\Phi^R_{t,t'}(i,j,k) = p(s_t = i, s_{t+1} = j|o_1, \ldots, o_{t'-1}, s_{t'} = k), \tag{26}$$

where the superscript R indicates $t' \geq t$.

The computation of ψ is initialized at $t' = t$ with

$$\psi_{t,t}(i,j) = \frac{\delta(i,j)}{b_i(o_t)}, \tag{27}$$

and we can compute the rest of the ψ iteratively as follows:

$$\psi^L_{t,t'-1}(i,j) = \sum_k \psi^L_{t,t'}(i,k) a_{jk} b_k(o_{t'}), \tag{28}$$

$$\psi^R_{t,t'+1}(i,j) = \sum_k \psi^R_{t,t'}(i,k) a_{kj} b_k(o_{t'}). \tag{29}$$

Finally, ϕ and Φ can be calculated based on ψ as follows:

$$\phi^L_{t,t'}(i,j) = \psi_{t,t'}(i,j) b_i(o_t) \frac{\beta_t(i)}{\beta_{t'}(j)}, \tag{30}$$

$$\phi^R_{t,t'}(i,j) = \psi_{t,t'}(i,j) b_j(o_{t'}) \frac{\alpha_t(i)}{\alpha_{t'}(j)}, \tag{31}$$

$$\Phi^L_{t,t'}(i,j,k) = \phi^L_{t+1,t}(j,i) \phi^L_{t,t'}(i,k), \tag{32}$$

$$\Phi^R_{t,t'}(i,j,k) = \phi^R_{t,t+1}(i,j) \phi^R_{t+1,t'}(j,k), \tag{33}$$

where $\alpha_t(i)$ and $\beta_t(i)$ are defined the same as in [8], and they can be calculated efficiently via the standard *forward-backward* algorithm.

After computing of all the values of ψ, ϕ, and Φ, we remove the superscript L and R on them since the subscripts t and t' have already encoded this information.

APPENDIX II
COMPUTATION OF THE JACOBIAN MATRICES

To compute the Jacobian matrices $\nabla_{\tilde{\Theta}} EM_0(\Theta)$ and $\nabla_{\tilde{\Theta}} EM_1(\Theta)$, we need to compute the gradients of $p(y|\boldsymbol{x}, \Theta)$, $\gamma_t(i)$ and $\xi_t(i,j)$ with respect to the parameters of the HMM classifier $\tilde{\Theta}$. This in turn requires the partial derivatives of $\log p(\boldsymbol{x}|\theta)$, $\gamma_t(i)$ and $\xi_t(i,j)$ with respect to each element of $\theta = \{\pi^N, A^{N\times N}, B^{N\times M}\}$. The computational formulas are specified as follows:

$$\begin{aligned}
\frac{\partial \log p(\boldsymbol{x}|\theta)}{\partial \pi_i} &= \frac{1}{p(\boldsymbol{x}|\theta)} \cdot \sum_{s_1,\cdots,s_T} \left[\frac{\partial \pi_{s_1}}{\partial \pi_i} \prod_{t=1}^{T-1} a_{s_t s_{t+1}} \prod_{t=1}^{T} b_{s_t}(x_t) \right] \\
&= \frac{1}{p(\boldsymbol{x}|\theta)} \cdot \sum_{s_1,\cdots,s_T} \left[\delta(s_1 = i) \prod_{t=1}^{T-1} a_{s_t s_{t+1}} \prod_{t=1}^{T} b_{s_t}(x_t) \right] \\
&= \frac{1}{p(\boldsymbol{x}|\theta)} \cdot \sum_{s_1} \delta(s_1 = i) b_{s_1}(x_1) \sum_{s_2,\cdots,s_T} \left[\prod_{t=1}^{T-1} a_{s_t s_{t+1}} \prod_{t=2}^{T} b_{s_t}(x_t) \right] \\
&= \frac{1}{p(\boldsymbol{x}|\theta)} \cdot b_i(x_1) \beta_1(i) = \gamma_1(i)/\pi_i,
\end{aligned} \tag{34}$$

$$\begin{aligned}
\frac{\partial \log p(\boldsymbol{x}|\theta)}{\partial a_{ij}} &= \frac{1}{p(\boldsymbol{x}|\theta)} \cdot \sum_{s_1,\cdots,s_T} \left[\pi_{s_1} \prod_{t=1}^{T} b_{s_t}(x_t) \frac{\partial}{\partial a_{ij}} \prod_{t=1}^{T-1} a_{s_t s_{t+1}} \right] \\
&= \frac{1}{p(\boldsymbol{x}|\theta)} \cdot \sum_{s_1,\cdots,s_T} \left[\pi_{s_1} \prod_{t=1}^{T} b_{s_t}(x_t) \sum_{k=1}^{T-1} \left(\delta(s_k = i, s_{k+1} = j) \prod_{t=1, t\neq k}^{T-1} a_{s_t s_{t+1}} \right) \right] \\
&= \frac{1}{p(\boldsymbol{x}|\theta)} \cdot \sum_{k=1}^{T-1} \left[\sum_{s_1,\cdots,s_T} \left(\pi_{s_1} \prod_{t=1}^{T} b_{s_t}(x_t) \delta(s_k = i, s_{k+1} = j) \prod_{t=1, t\neq k}^{T-1} a_{s_t s_{t+1}} \right) \right] \\
&= \frac{1}{p(\boldsymbol{x}|\theta)} \cdot \sum_{t=1}^{T-1} \alpha_t(i) b_j(x_{t+1}) \beta_{t+1}(j) = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{a_{ij}}, \qquad (35)
\end{aligned}$$

$$\frac{\partial \log p(\boldsymbol{x}|\theta)}{\partial b_i(v)} = \frac{1}{p(\boldsymbol{x}|\theta)} \cdot \sum_{t=1}^{T} \frac{\alpha_t(i)\beta_t(i)}{b_i(x_t)} \delta(x_t = v) = \frac{\sum_{t=1}^{T} \gamma_t(i) \delta(x_t = v)}{b_i(v)}. \qquad (36)$$

Similarly, the partial derivatives of $\gamma_t(i)$ and $\xi_t(i,j)$ can be calculated as

$$\frac{\partial \gamma_t(i)}{\partial \pi_k} = \frac{\gamma_t(i)}{\pi_k} \left[\phi_{1,t}(k,i) - \gamma_1(k) \right], \qquad (37)$$

$$\frac{\partial \gamma_t(i)}{\partial a_{kl}} = \frac{\gamma_t(i)}{a_{kl}} \left[\sum_{t'=1}^{T-1} \Phi_{t',t}(k,l,i) - \sum_{t'=1}^{T-1} \xi_{t'}(k,l) \right], \qquad (38)$$

$$\frac{\partial \gamma_t(i)}{\partial b_k(v)} = \frac{\gamma_t(i)}{b_k(v)} \left[\sum_{t'=1}^{T} (\phi_{t',t}(k,i) - \gamma_{t'}(k)) \delta(o_{t'} = v) \right], \qquad (39)$$

$$\frac{\partial \xi_t(i,j)}{\partial \pi_k} = \frac{\xi_t(i,j)}{\pi_k} \left[\phi_{1,t}(k,i) - \gamma_1(k) \right], \qquad (40)$$

$$\frac{\partial \xi_t(i,j)}{\partial a_{kl}} = \frac{\xi_t(i,j)}{a_{kl}} \left[\sum_{t'=1}^{t-1} \Phi_{t',t}(k,l,i) + \delta(k=i, l=j) + \sum_{t'=t+1}^{T-1} \Phi_{t',t+1}(k,l,j) - \sum_{t'=1}^{T-1} \xi_{t'}(k,l) \right], \qquad (41)$$

$$\frac{\partial \xi_t(i,j)}{\partial b_k(v)} = \frac{\xi_t(i,j)}{b_k(v)} \left[\sum_{t'=1}^{t} \phi_{t',t}(k,i) \delta(o_{t'} = v) + \sum_{t'=t+1}^{T} \phi_{t',t+1}(k,j) \delta(o_{t'} = v) - \sum_{t'=1}^{T} \gamma_{t'}(k) \delta(o_{t'} = v) \right] \qquad (42)$$

By the chain rule, the partial derivatives with respect to each element of $\tilde{\Theta}$ can be calculated as, for example,

$$\frac{\partial \gamma_t(i)}{\partial \tilde{\pi}_k} = \sum_j \frac{\partial \gamma_t(i)}{\partial \pi_j} \cdot \frac{\partial \pi_j}{\partial \tilde{\pi}_k} = \left(\frac{\partial \gamma_t(i)}{\partial \pi_k} - \sum_j \pi_j \frac{\partial \gamma_t(i)}{\partial \pi_j} \right) / \sum_j \tilde{\pi}_j, \qquad (43)$$

and similarly for $\tilde{a}_{kl}$ and $\tilde{b}_k(v)$.

Finally, the partial derivatives of $p(y|\boldsymbol{x},\Theta)$ with respect to each element of $\tilde{\Theta}$ are given as follows:

$$\frac{\partial p(y|\boldsymbol{x},\Theta)}{\partial \tilde{w}^{y'}} = \varepsilon/\tilde{w}^{y'}, \tag{44}$$

$$\frac{\partial p(y|\boldsymbol{x},\Theta)}{\partial \tilde{\pi}_k^{y'}} = \varepsilon/\tilde{\pi}_k^{y'} \cdot [\gamma_1^{y'}(k) - \pi_k^{y'}], \tag{45}$$

$$\frac{\partial p(y|\boldsymbol{x},\Theta)}{\partial \tilde{a}_{kl}^{y'}} = \varepsilon/\tilde{a}_{kl}^{y'} \cdot \left[\sum_{t=1}^{T-1} \xi_t^{y'}(k,l) - a_{kl}^{y'} \sum_{t=1}^{T-1} \gamma_t^{y'}(k)\right], \tag{46}$$

$$\frac{\partial p(y|\boldsymbol{x},\Theta)}{\partial \tilde{b}_k^{y'}(v)} = \varepsilon/\tilde{b}_k^{y'}(v) \cdot \left[\sum_{t=1}^{T} \gamma_t^{y'}(k)\delta(o_t{=}v) - b_k^{y'}(v) \sum_{t=1}^{T} \gamma_t^{y'}(k)\right], \tag{47}$$

where

$$\varepsilon = p(y'|\boldsymbol{x},\Theta) \cdot [\delta(y{=}y') - p(y|\boldsymbol{x},\Theta)]\,. \tag{48}$$