Pattern Recognition Letters 37 (2014) 161–171

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Joint semi-supervised learning of Hidden Conditional Random Fields and Hidden Markov Models

Yann Soullard [*], Martin Saveski, Thierry Artières

LIP6, Université Pierre et Marie Curie, 4 place Jussieu, 75005 Paris, France

ARTICLE INFO

Article history:
Available online 6 April 2013

Keywords:
Hidden Markov Models
Hidden Conditional Random Fields
Semi-supervised learning
Co-training

ABSTRACT

Although semi-supervised learning has generated great interest for designing classifiers on static patterns, there has been comparatively fewer works on semi-supervised learning for structured outputs and in particular for sequences. We investigate semi-supervised approaches for learning hidden state conditional random fields for sequence classification. We propose a new approach that iteratively learns a pair of discriminative-generative models, namely Hidden Markov Models (HMMs) and Hidden Conditional Random Fields (HCRFs). Our method builds on simple strategies for semi-supervised learning of HMMs and on strategies for initializing HCRFs from HMMs. We investigate the behavior of the method on artificial data and provide experimental results for two real problems, handwritten character recognition and financial chart pattern recognition. We compare our approach with state of the art semi-supervised methods.

1. Introduction

Sequence classification and sequence labeling are fundamental tasks occurring in many application domains, such as speech recognition, mining financial time series, and handwriting recognition. Hidden Markov Models (HMMs) are the most popular method for dealing with sequential data (Rabiner, 1989). HMMs benefit from efficient algorithms both for inference and for training but suffer some severe drawbacks. In particular, they are traditionally learned via maximum likelihood estimation, which is a non discriminative training criterion. Many attempts have been made to overcome this limitation, relying on the optimization of a discriminant criterion like minimum error classification (Juang and Katagiri, 1992), perceptron loss (Collins, 2002), maximum mutual information (Woodland and Povey, 2002), or margin-based criterion (Sha and Saul, 2007; Do and Artières, 2009). A more recent alternative consists in defining a model of the posterior conditional probability (i.e. the probability of the labeling given the observation sequence). Hidden Conditional Random Fields (HCRFs) are such models (Quattoni et al., 2007). They are a variant of Conditional Random Fields (CRFs) (Lafferty et al., 2001) that make use of hidden states to account for the underlying structure of the data (alike in HMMs). They have been used for various signal labeling tasks, in particular for speech signals (Gunawardana et al., 2005; Reiter et al., 2007), eye movements (Do and Artières, 2005), handwriting (Do and Artières, 2006; Vinel et a., 2011), gestures and images (Morency et al., 2007) and financial time series (Soullard and Artieres, 2011).

Whatever the model one chooses to design a classification system, one needs first to gather, then to label, a sufficiently large training corpus. This often has a cost that may make the design of a good system problematic. This has motivated the study of semi-supervised learning (SSL). In SSL, classifiers are trained on both labeled samples (usually few) and unlabeled samples (usually many). A number of SSL methods have been proposed, such as entropy based methods (Grandvalet and Bengio, 2005), margin based methods (Wang et al., 2009), co-training algorithms (Blum and Mitchell, 1998) (see Mann and McCallum, 2010 for a review).

However, up to now only a few works have investigated semi-supervised learning for structured data and for sequences in particular, as we are interested in here. Some studies have investigated semi-supervised learning of HMMs for speech recognition and for text classification (Nigam et al., 2000; Inoue and Ueda, 2003; Haffari and Sarkar, 2008), but the conclusions of these works are rather limited since SSL has been shown to eventually degrade performances of supervised training (Cozman and Cohen, 2002; Mérialdo, 1994). Moreover, alternative works have focused on learning CRFs in a semi-supervised setting for language processing and biological problems, yielding some significant improvements (Jiao, 2006; Sokolovska, 2011). It is worth noting that a few of these works rely on designing a hybrid model, mixing HMMs and CRFs, where HMMs only are learned in a semi-supervised way, indirectly making the learning of CRFs semi-supervised (Sokolovska, 2011). Finally, we are not aware of any work today on SSL algorithms for complex discriminative models such as HCRFs.

[*] Corresponding author. Tel.: +33 1 44 27 74 91; fax: +33 1 44 27 70 00.
E-mail addresses: Yann.Soullard@lip6.fr (Y. Soullard), Martin.Saveski@lip6.fr (M. Saveski), Thierry.Artieres@lip6.fr (T. Artières).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.03.028

Here we focus on semi-supervised learning for sequence classification where one wants to assign a single label to an input sequence. Extension to sequence labeling is out of the scope of the paper but should follow naturally. We propose a new algorithm for semi-supervised learning of HCRFs for sequence classification. It relies on an iterative joint learning of a pair of generative and discriminative models, namely HMMs and HCRFs. This paper is an extension of our previous work in Soullard and Artieres (2011), and improves on it in several ways. First, we describe in more detail our approach, in particular the initialization scheme of HCRF from Full Covariance matrix Gaussian HMMs. Second, we propose and investigate a few variants of our method. Third, we provide new results on artificial data for an improved understanding of the behavior of the method. Fourth, we provide additional results on real datasets and provide a thorough experimental comparison of our approach with state of the art SSL methods that were already proposed for CRFs and that we extended to HCRFs.

We first present related works on semi-supervised learning in Section 2, then we detail in Section 3 our strategy for initializing HCRFs from Full Covariance matrix Gaussian HMMs. Next, we discuss the motivation of our approach, which we present in detail in Section 4. We report experimental results on artificial data in Section 5 and we investigate in Section 6 the behavior of our approach for two real problems, handwritten character recognition and financial chart pattern classification.

2. State of the art in semi-supervised learning

Here, we review the main semi-supervised learning approaches (Zhu and Goldberg, 2009), with a particular focus on methods that have been used or that could be extended for learning markovian models such as HMMs and CRFs.

In this study, we focus on classification where training samples are couples $(\mathbf{x}, y), \mathbf{x} \in \mathcal{X}$ is an input sample (e.g. a sequence) and where $y \in \mathcal{Y}$ is its class (i.e. label).[1] We denote $L = \{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^{|L|}, y^{|L|})\}$ as the set of labeled training samples, with $|L|$ being its cardinal, and $U = \{\mathbf{x}^{|L|+1}, \ldots, \mathbf{x}^{|L|+|U|}\}$ stands for the set of unlabeled training samples. Also, in the following we will systematically use Θ to denote the set of parameters of generative models (e.g. HMMs) and Λ to denote the set of parameters of discriminative models (e.g. CRFs).

2.1. Mixture approach

The *mixture approach* consists of learning a mixture of generative models, one for each class, through an Expectation Maximization (EM) like algorithm. In Nigam et al. (2000), the EM algorithm was applied on a mixture of multinomial distributions for text classification while in Baluja (1998) it was applied on a face orientation discrimination task. This approach has been applied to HMMs in Nigam et al. (2000); Inoue and Ueda (2003). The objective criterion to be maximized is defined as:

$$\mathcal{L}(\Theta) = \frac{(1-\gamma)}{|L|}\sum_{i=1}^{|L|} log\, p(\mathbf{x}^{(i)}, y^{(i)}|\Theta) + \frac{\gamma}{|U|}\sum_{j=|L|+1}^{|L|+|U|} log\, p(\mathbf{x}^{(j)}|\Theta) \quad (1)$$

where $\gamma \in [0, 1]$ is a parameter that allows tuning of the relative influence of labeled data and unlabeled data. The fully supervised and the fully unsupervised cases are specific instances when γ is respectively set to 0 and to 1 (Ji et al., 2009). Although it is a simple and attractive idea, this approach may degrade HMMs' performances (Cozman and Cohen, 2002; Mérialdo, 1994), especially if the number of labeled samples is too small.

2.2. Minimum entropy

Minimum entropy regularization is a popular technique (Grandvalet and Bengio, 2005). It aims at reducing uncertainty on the labeling of unlabeled samples. the method is extended in Jiao (2006) to the learning of CRF and is used with the following regularized objective function:

$$\mathcal{L}_\gamma(\Lambda) = -\frac{||\Lambda||^2}{2} + \sum_{i=1}^{|L|} log\, p(y^{(i)}|\mathbf{x}^{(i)}, \Lambda) + \gamma \sum_{j=|L|+1}^{|L|+|U|} \sum_{y \in \mathcal{Y}} p(y|\mathbf{x}^{(j)}, \Lambda) log\, p(y|\mathbf{x}^{(j)}, \Lambda) \quad (2)$$

The above objective combines conditional entropy for unlabeled samples and conditional likelihood for labeled samples. A similar approach was taken in Wang et al. (2009) by defining the objective function as the combination of the conditional likelihood of the labeled data and of the mutual information for the unlabeled data.

2.3. Co-training

Co-training has been popularized by Blum and Mitchell (1998) for static patterns. It assumes that the features used to represent a sample may be split into two sets of features, or views, (every sample then has two representations, one for each view) and that these two views are sufficient for a correct classification. Learning consists of first training two classifiers, one for each view. Then one selects the unlabeled samples for which one classifier is most confident and puts these samples together with the classifier's predictions into the training set of the other classifier. This process is repeated iteratively. The approach is extended in Wang and Zhou (2007) to the case where two classifiers are trained on the same view and showed that co-training may work well provided the classifiers are different enough.

Co-training has also been investigated with some success for learning generative markovian models. In particular, the standard co-training algorithm was applied in Khine et al. (2008) to HMMs for singing voice detection and co-training of HMMs and of neural networks was experimented in Frinken et al. (2009) for handwriting recognition.

2.4. Hybrid methods

A few methods have been proposed to mix generative and discriminative methods (Bishop and Lasserre, 2007; Bouchard, 2007). These methods rely on the idea that semi-supervised learning is more natural for learning generative models with a non discriminative criterion through, e.g. the *mixture approach*. In Bouchard (2007), the parameters of generative models are learnt by optimizing a combination of a non discriminative criterion (e.g. likelihood) and of a discriminative criterion (conditional likelihood), where the non discriminative criterion is computed for all training data (labeled and unlabeled) while the discriminative criterion concerns labeled training data only. Furthermore, some authors proposed in Bishop and Lasserre (2007) to learn two linked sets of parameters of generative models, one parameter set with the non discriminative criterion (on the entire training dataset) and the other parameter set with the discriminative criterion (on the labeled training dataset) with the following objective function:

$$\mathcal{L}(\Theta, \Lambda) = \sum_{i=1}^{|L|} \log p(y^{(i)}|\mathbf{x}^{(i)}, \Lambda) + \sum_{j=1}^{|L|+|U|} \log p(\mathbf{x}^{(j)}|\Theta) + log(p(\Theta, \Lambda)) \quad (3)$$

where $p(\Theta, \Lambda)$ is a prior that links the two parameter sets. It allows blending generative and discriminative approaches. If the prior is uniform, the generative and discriminative models are

[1] Note that we use bold font to denote sequences, e.g. $\mathbf{x}$, while we use normal font for static patterns, vector or scalar, e.g. y.

independently trained so that the discriminative model is learned in a fully supervised setting. If the prior forces Θ and Λ to coincide, the two models are constrained and this reduces to the approach in Minka (2005). In addition, if the prior is smoother (e.g. $e^{\|\Theta-\Lambda\|^2}$), the discriminative model is learned in a supervised setting with the constraint of being not too far from the generative model, which is learned in a semi-supervised way.

Furthermore, cascading models was investigated in Suzuki et al. (2007). The outputs produced by a few discriminative (CRFs) and generative models (HMMs) was combined in a CRF-like model, where generative models are learned in a semi-supervised setting while discriminative models are learned in a supervised mode. Finally, a principled and asymptotically optimal way to train a CRF with a weighted conditional likelihood was proposed in Sokolovska (2011). The weight of a particular training sequence is the estimated density for this sequence, which is approximated by generative models trained in a semi-supervised setting.

3. Initialization of HCRFs from HMMs

In the following we consider that an input sample $\mathbf{x}$ is a sequence of length T (sequences are noted in bold) $\mathbf{x} = (x_1, \ldots, x_T) \in \mathcal{X}$ whose tth element (named a frame) $x_t \in \mathbb{R}^d$ is a d-dimensional feature vector.

One difficulty for learning HCRFs, unlike CRFs, lies in the non convexity of the training criterion which comes from the introduction of hidden states. This makes training sensitive to initialization and easily leads to overfitting. A solution has been proposed in Gunawardana et al. (2005) to overcome this problem. It consists of learning first a HMM system (e.g. one left right HMM per class in our case) through Maximum Likelihood Estimation (MLE) and then of initializing a HCRF with parameter values such that the obtained HCRF behaves exactly as the HMM system does (note that the HCRF must share the same topology as the HMM system, hence it is composed of many left–right chains, one for every class).

Although the learning criterion of HMMs (e.g. Maximum Likelihood) is also non convex, HMMs trained with Maximum Likelihood Estimation are usually less subject to overfitting than HCRFs for two reasons. First, HMMs are less powerful models than HCRFs for classification (as this discussion will show one can design a HCRF system that behaves just as a given HMM system does while the reciprocal is not true). Second, non discriminative models (e.g. HMMs trained via MLE) are usually less subject to overfitting than discriminatively trained models (Bouchard and Triggs, 2004).

We briefly explain now how to initialize a HCRF system from a HMM system in the case of single Gaussian HMMs with a full covariance matrix (the works in Gunawardana et al. (2005) dealt with the diagonal covariance matrix case).

The key point is that the log of the joint probability of an input sequence (i.e. an observation sequence) and of a state sequence may be written as a dot product between a parameter vector and a joint feature map depending on the sequence of hidden states and on the observation sequence. We explain this now. The log joint probability of an observation sequence $\mathbf{x}$ and of a state sequence $\mathbf{h}$ may be written as (following Rabiner, 1989):

$$\log p(\mathbf{x}, \mathbf{h}|\Theta) = \log(\pi_{h_1}) + \log(p(x_1|h_1)) + \sum_{t=2}^{T} (\log p(h_t|h_{t-1}) + \log p(x_t|h_t, \Theta)) \quad (4)$$

where x_t denotes the tth frame of $\mathbf{x}$, h_t stands for the state at time t (with $h_t \in [1, Q] \forall t$), $p(x_t|q, \Theta)$ stands for emission probability in state q, which is a Gaussian distribution with mean $\mu^q \in \mathbb{R}^d$ and covariance matrix Σ^q, whose determinant will be denoted $|\Sigma^q|$. Finally, we will denote μ^{h_t} and Σ^{h_t} as the mean and the covariance matrix of the distribution in state h_t.

First, note that the first term in Eq. 4, $\log(\pi_{h_1})$ may be simply written as:

$$\log(\pi_{h_1}) = \begin{pmatrix} \log \pi_1 \\ \ldots \\ \log \pi_Q \end{pmatrix} . \begin{pmatrix} \delta_{[h_1=1]} \\ \ldots \\ \delta_{[h_1=Q]} \end{pmatrix}$$

where $\delta_{[P]}$ equals 1 if predicate P is true and 0 otherwise, and where $u.v$ denotes the dot product between two vectors u and v.

Hence $\log(\pi_{h_1})$ may be put in the shape of a dot product of a parameter vector (left vector in the right hand side of the previous equation) and of a feature vector computed from $\mathbf{x}$ and $\mathbf{h}$ (right vector). Similarly, one can write, using the usual notation $a_{i,j} = p(h_t = j \mid h_{t-1} = i)$:

$$\log a_{h_{t-1},h_t} = \begin{pmatrix} \log a_{1,1} \\ \log a_{1,2} \\ \ldots \\ \log a_{Q,Q} \end{pmatrix} . \begin{pmatrix} \delta_{[h_{t-1}=1 \text{ and } h_t=1]} \\ \delta_{[h_{t-1}=1 \text{ and } h_t=2]} \\ \ldots \\ \delta_{[h_{t-1}=Q \text{ and } h_t=Q]} \end{pmatrix}$$

Now let $\mathbf{w}^{trans}$ denote the left vector of the right hand side in the previous equation and let $\phi^{trans}(\mathbf{x}, \mathbf{h}, t)$denote the right vector[2]:

$$\mathbf{w}^{trans} = (\log a_{1,1}, \log a_{1,2} \ldots, \log a_{Q,Q})'$$

$$\phi^{trans}(\mathbf{x}, \mathbf{h}, t) = \left(\delta_{[h_{t-1}=1\text{and}h_t=1]}, \delta_{[h_{t-1}=1\text{and}h_t=2]}, \ldots, \delta_{[h_{t-1}=Q\text{and}h_t=Q]}\right)'$$

Then one can also put the sum of probability transition terms in Eq. 4 in the shape of a dot product:

$$\sum_{t=2}^{T} \log p(h_t|h_{t-1}) = \mathbf{w}^{trans}.\Phi^{trans}(\mathbf{x}, \mathbf{h}) \quad (5)$$

with $\Phi^{trans}(\mathbf{x}, \mathbf{h}) = \sum_{t=2}^{T} \phi^{trans}(\mathbf{x}, \mathbf{h}, t)$.

The emission log probability of a frame x in a state q, $\log p(x|q, \Theta)$ may also be written as the dot product between two vectors, a parameter vector built from the parameters of the Gaussian distribution in state q (with parameters μ, Σ) and a feature vector built from x. First, note that:

$$\log p(x|q, \Theta) = -\frac{1}{2}\left(x'\Sigma^{-1}x - x'\Sigma^{-1}\mu - \mu'\Sigma^{-1}x + \mu'\Sigma^{-1}\mu + \log((2\pi)^d|\Sigma|)\right)$$

Hence:

$$p(x|q, \Theta) = \begin{pmatrix} \mu'\Sigma^{-1}\mu - \log((2\pi)^d|\Sigma|) \\ \left(-2\mu'\Sigma^{-1}\right) \\ Vec(\Sigma^{-1}) \end{pmatrix} . \begin{pmatrix} 1 \\ x \\ Vec(x \otimes x) \end{pmatrix}$$

where $Vec(\Sigma^{-1})$ stands for the matrix Σ^{-1} (whose elements are noted $\tilde{\sigma}_{p,q}$, i.e. $\Sigma^{-1} = \{\tilde{\sigma}_{k_1,k_2}\}_{k_1,k_2=1..d}$) that has been put in a vector shape: $Vec(\Sigma^{-1}) = (\tilde{\sigma}_{1,1}, \tilde{\sigma}_{2,1}, \ldots, \tilde{\sigma}_{1,2}, \ldots, \tilde{\sigma}_{d,d})'$, and $Vec(x \otimes x)$ stands for the tensor product $x \otimes x = (x_1^2, x_2x_1, \ldots, x_2^2, \ldots, x_d^2)'$ put in a vector shape. More generally, denoting w_q^{pdf} as the parameter vector for state q (i.e. $w_q^{pdf} = \left(\mu^{q\prime}\Sigma^{q-1}\mu^q - \log((2\pi)^d|\Sigma^q|), \left(-2\mu^{q\prime}\Sigma^{q-1}\right)', Vec(\Sigma^{q-1})'\right)'$, and denoting $\tilde{x} = (1, x', (Vec(x \otimes x))')'$, one may use the following form to compute $\log p(x_t|h_t, \Theta)$:

$$\log p(x_t|h_t, \Theta) = \begin{pmatrix} w_1^{pdf} \\ \ldots \\ w_Q^{pdf} \end{pmatrix} . \begin{pmatrix} (\delta_{h_t=1} \times \tilde{x}) \\ (\delta_{h_t=2} \times \tilde{x}) \\ \ldots \\ (\delta_{h_t=Q} \times \tilde{x}) \end{pmatrix}$$

[2] We use u' to denote the transpose of vector u.

Then denoting w^{pdf} as the left vector in the right hand side of previous equation and $\phi^{pdf}(\mathbf{x},\mathbf{h},t)$ the right vector in the right hand side, one clearly sees that:

$$\sum_{t=1}^{T} \log p(x_t|h_t,\Theta) = w^{pdf}.\Phi^{pdf}(\mathbf{x},\mathbf{h}) \quad (6)$$

with $\Phi^{pdf}(\mathbf{x},\mathbf{h}) = \sum_{t=1}^{T}\phi^{pdf}(\mathbf{x},\mathbf{h},t)$.

At the end, by concatenating the parameter vectors for the initial state probability, the transition probabilities and the pdfs into a parameter vector $\mathbf{w}_\Theta$, and concatenating the corresponding features vectors into a global feature vector $\Phi(\mathbf{x},\mathbf{h})$, one can see that:

$$\log p(\mathbf{x},\mathbf{h}|\Theta) = \mathbf{w}_\Theta.\ \Phi(\mathbf{x},\mathbf{h})$$

Furthermore using above result, one may write:

$$p(\mathbf{h}|\mathbf{x},\Theta) = \frac{p(\mathbf{x},\mathbf{h}|\Theta)}{\sum_{\mathbf{k}} p(\mathbf{x},\mathbf{k}|\Theta)} = \frac{e^{\mathbf{w}_\Theta.\Phi(\mathbf{x},\mathbf{h})}}{\sum_{\mathbf{k}} e^{\mathbf{w}_\Theta.\Phi(\mathbf{x},\mathbf{k})}}$$

The above result yields an efficient learning procedure for learning HCRFs. Indeed the posterior probability of a state sequence given an observation sequence computed by a HCRF with parameters Λ is expressed with a similar shape:

$$p(\mathbf{h}|\mathbf{x},\Lambda) = \frac{e^{\mathbf{w}_\Lambda.\Phi(\mathbf{x},\mathbf{h})}}{\sum_{\mathbf{k}} e^{\mathbf{w}_\Lambda.\Phi(\mathbf{x},\mathbf{k})}} \quad (7)$$

where the denominator is called the partition function. Note that in HCRF the joint feature map Φ is decomposable over the cliques of the dependency graph of random variables (i.e. states) (Quattoni et al., 2007). Although computing the partition function may be very costly and may become a potentially serious complexity problem in the general case, here we focus on small-cliques (h_{t-1},h_t) and (h_t,x_t) which make the computation feasible in an efficient way. This is a common strategy when using hidden CRF for signals since it yields an efficient computation of the partition function through dynamic programming routines that are similar to the forward backward algorithm for HMMs (Gunawardana et al., 2005Reiter et al., 2007Do and Artières, 2006).

It is now clear that a HMM system may be transformed into a HCRF system by using the above definitions for $\mathbf{w}$ and Φ. Assume that a learned HMM system (with one HMM per class) is available. It may be considered a big HMM where a particular state sequence translates into a class label. Then one can initialize a HCRF system with the same topology as this big HMM using the above formulas. By construction, this HCRF outputs exactly the same classification decision as the HMM system.

One can then optimize from this initial solution the standard discriminative conditional likelihood criterion of HCRFs to fine-tune the HCRF system. One may expect that such an initialization by the HMM system allows starting the HCRF optimization process in an interesting area so as to reach a relevant local minimum of the non convex HCRF optimization criterion.

4. Joint semi-supervised learning for HMMs and HCRFs

Designing semi-supervised learning algorithms for HCRF is not straightforward. A first solution is to extend traditional SSL approaches to HCRFs which we will investigate in the experimental section. Starting from general ideas on the performance of generative and discriminative systems for classification we chose instead to design a new approach where we jointly learn iteratively a HMM and a HCRF system. We first discuss the motivation of this work, then we present in detail our method and a few variants.

4.1. Motivation

The starting point of our approach lies in general observations concerning the training and generalization ability of non discriminative and of discriminative approaches with small training datasets.

On the one hand, non discriminative approaches (e.g. HMMs trained through MLE) rely on the learning of one model per class and estimate a (class conditional) distribution over observations $p(\mathbf{x} \mid y)$. As suggested in Bouchard and Triggs (2004), these approaches may exhibit a lower variance than discriminative models, focusing on $p(y \mid \mathbf{x})$, but may have a higher bias unless the parametric model one chooses for estimating $p(\mathbf{x} \mid y)$ is an accurate model of the actual distribution (which is wrong in general when using HMMs for instance). On the other hand, the discriminative approach (e.g. HCRFs) focuses on modeling the posterior distribution which is directly related to the classification goal, but it usually comes with more powerful models that are more subject to overfitting for small datasets. Furthermore, authors investigated in Ng and Jordan (2001) a particular pair of generative-discriminative models: Naive Bayes and Logistic Regression models. They showed on the one hand that the discriminative model has a lower asymptotic error than the generative model (as the number of training samples becomes large) and on the other hand that the generative model may approach its asymptotic error much faster than the discriminative model. As a consequence, it may happen that generative models may be more accurate with a small training dataset while discriminative models outperform generative ones when the training set size increases. Although the relevance of such general comments is questionable, it definitely suggests that mixing both approaches is appealing when the training set size is small, as the case in the semi-supervised setting.

4.2. Iterative Hybrid Algorithm (IHA)

We present now our algorithm which we call the Iterative Hybrid Algorithm (IHA), it is illustrated in Fig. 1. It starts by training an initial HMM system on labeled and on unlabeled data $(L \cup U)$ with a semi-supervised learning algorithm such as the *mixture* approach. Then the algorithm iterates a two step process, where we first train a discriminative HCRF system using the current HMM system and second we retrain the HMM system using the outputs produced by the HCRF system.

To train the HCRF system, we first initialize it from the HMM system using the strategy described in Section 3. Then the HCRF

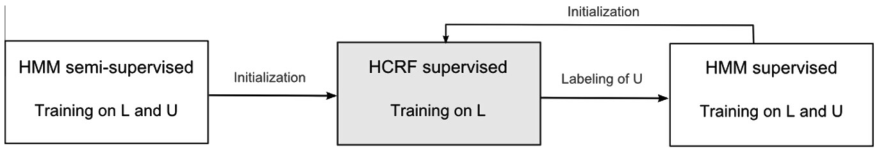

Fig. 1. Semi-supervised strategy embedding HMM and HCRF learning, L and U denote the sets of labeled and unlabeled training sequences.

is retrained on L with a regularization term that constrains the solution to lie close to the HMM solution it is initialized from. Then, we retrain the HMM system. To do this, we use the HCRF system to label part (or all) of the unlabeled data U and we use it together with L to retrain a HMM system in a supervised mode. We repeat this process for a number of iterations, or until convergence.

More formally, the IHA algorithm consists of the following steps:

1. Initialization
(a) Semi-supervised learning of Θ on $L \cup U$ yielding $\Theta^{(0)}$:

$$\Theta^{(0)} = argmax_{\Theta}\left(\frac{\gamma}{|L|}\sum_{i=1}^{|L|}\log p(\mathbf{x}^{(i)}, y^{(i)}|\Theta) + \frac{(1-\gamma)}{|U|}\sum_{j=|L|+1}^{|L|+|U|}\log\sum_{y'\in\mathcal{Y}}p(\mathbf{x}^{(j)}, y'|\Theta)\right) \tag{8}$$

(b) k = 1
2. Loop for a fixed number of iterations or until convergence. At iteration k:
   (a) Initialization of $\tilde{\Lambda}^{(k)}$, starting from $\Theta^{(k-1)}$ (cf. Section 3).
   (b) Supervised Learning of Λ on L via Stochastic Gradient Descent on the following objective function, starting from $\tilde{\Lambda}^{(k)}$:

$$\Lambda^{(k)} = argmax_{\Lambda}\sum_{i=1}^{|L|}\log p(y^{(i)}|\mathbf{x}^{(i)}, \Lambda) - \frac{1}{2}\|\Lambda - \Theta^{(k-1)}\|^2 \tag{9}$$

(c) Use $\Lambda^{(k)}$ to label part of U which becomes $U_{Labeled}$, where the labels are assigned as:

$$\forall j \in [|L|+1, |L|+|U|], \hat{y}^{(j)} = argmax_{y\in\mathcal{Y}}\, p(y|\mathbf{x}^{(j)}, \Lambda^{(k)}) \tag{10}$$

(d) Supervised Learning of Θ on $L + U_{Labeled}$ yielding $\Theta^{(k)}$:

$$\Theta^{(k)} = argmax_{\Theta}\left(\frac{\gamma}{|L|}\sum_{i=1}^{|L|}\log p(\mathbf{x}^{(i)}, y^{(i)}|\Theta) + \frac{(1-\gamma)}{|U|}\sum_{j=|L|+1}^{|L|+|U_{Labeled}|}p(\hat{y}^{(j)}|\mathbf{x}^{(j)}, \Lambda)\log p(\mathbf{x}^{(j)}, \hat{y}^{(j)}|\Theta)\right) \tag{11}$$

(e) k = k + 1; Goto 3.1

The only hyper parameter of the algorithm (except prior choices such as the topology of the models) is γ, which controls the influence of the unlabeled data U.

Note that the above algorithm may be used for learning any pair of generative and discriminative models provided there is a way for initializing the discriminative models from the generative ones (e.g. Naive Bayes and logistic regression on static patterns). In particular it may be used for learning two generative systems that share the same structure, where one system is trained with a non discriminative criterion and the second one is learned with a discriminative criterion.

Our method bares some similarities with the Segmental K-Means algorithm (Rabiner, 1989Juang and Rabiner, 1990) and with the pioneer work from Morgan and Bourlard (1995) on hybrid HMM/Neural Networks models. For instance this latter work focuses on an iterative framework where a neural net is trained first, then a hybrid HMM is used along with the neural net to perform a new (and better) segmentation (i.e., labeling) of the training sequences. Then, the neural net is trained again on the new labeling and the process is repeated.

4.3. Discussion

The first idea of the algorithm is to learn the discriminative model in a purely supervised way starting from a HMM that has been learned in a semi-supervised setting and regularizing around this initial solution. The HCRF solution at this step will be a local optimum of the regularized conditional likelihood criterion. Being optimized with a discriminative criterion one may expect the solution to be more accurate than the HMM it is initialized from. Being constrained to be not too far from the HMM solution, one expects the HCRF solution will indirectly take into account the unlabeled data and will be less subject to overfitting.

Secondly, the way the HCRF influences the HMM learning is close to the co-training idea which has been shown to be efficient in many situations (Blum and Mitchell, 1998,Wang and Zhou, 2007,Khine et al., 2008). Also the objective function for retraining the HMM system is close to a semi-supervised criterion as in the mixture approach, but where the weight of a sequence to reestimate the model of a class is given by the HCRF system. If the latter is more powerful than the HMM system, then one may expect it will provide better labels.

From this discussion, although we have no theoretical proof, we may expect at the end of any iteration that the HMM system will improve over the HMM system at the previous iteration. Since in the HCRF system one builds every iteration based on the HMM system and should improve over it, one can also expect that the successive HCRF systems will exhibit steadily improving performances.

4.4. Alternative strategies

We investigated a few variants for retraining the generative system in step 2.4 of the IHA algorithm. In any case the model of a class c is trained to maximize an objective function that is composed of the likelihood of the labeled samples for that class and of an additional term that depends on unlabeled data. The variants differ in the definition of this additional term.

In a first variant, every unlabeled sample is used to retrain every class model with a weight equal to its posterior probability, as given by the discriminative system. In this case, our algorithm comes close to the standard semi-supervised framework of generative models (see 2.1), but where the weight of a sequence for reestimating the model of class c is given by the HCRF system. The objective function for the learning model of class c, whose parameters are noted θ_c, is written as:

$$\begin{aligned}\mathcal{L}(\theta_c) &= \gamma\frac{1}{L_c}\sum_{i\in[1,|L|]\text{such that}y^{(i)}=c}\log p(\mathbf{x}^{(i)}, y^{(i)}\\ &= c|\theta_c) + (1-\gamma)\frac{1}{|U|}\sum_{j=|L|+1}^{|L|+|U|}p(y=c|\mathbf{x}^{(j)}, \Lambda)\log p(\mathbf{x}^{(j)}, y\\ &= c|\theta_c)\end{aligned} \tag{12}$$

We call this the *AllClasses* variant.

Second, an alternative consists in exploiting only unlabeled samples that would have been predicted in class c by the HCRF system to retrain the HMM of class c. We call this variant the *MaxProb* variant. Then the summation in the second term of Eq. 12 would concern only samples $j \in [|L|+1, |L|+|U|]$ such that $argmax_{c'}\,p(y = c'|\mathbf{x}^{(j)}, \Lambda) = c$.

Third, instead of weighting the contribution of the unlabeled samples by their posterior probability, as in Eq. (12), we may simply add samples to the HMM training set of their predicted classes according to the HCRF decision. In this case, the second sum of the above equation becomes $\sum_{j=|L|+1:|L|+|U|\text{s.t.}argmax_{c'}p(y=c'|\mathbf{x}^{(j)},\Lambda)=c}$ log $p(\mathbf{x}^{(j)}|\theta_c)$. We call this strategy the *WeightOne* variant.

Finally, a variant of the *WeightOne* case consists of considering only very likely samples whose conditional probability (given by the discriminative model) is over a threshold τ (e.g. close to one). This is related to a co-training strategy where a limited number of training samples, labeled by the discriminative system, would be added to the training set of the generative system. We call this strategy *SelectProb*.

5. Experiments on artificial data sets

Using synthetic data, we first compare the performances of the IHA and benchmark methods: the Entropy Minimization (Grandvalet and Bengio, 2005) (named EM hereafter) and the Hybrid Model from Bishop and Lasserre (2007); Lasserre (2008) (named HM hereafter).

We built a binary classification problem by generating two dimensional static patterns (i.e. samples are not sequences) with two Gaussian distributions, one for each class. Such simple data allows visual investigation. The class-conditional densities $p(x|y)$ have the same variance on the y-axis, but are horizontally elongated.

We investigated the abilities of semi-supervised approaches to learn one isotropic Gaussian distribution per class. This model does not capture the horizontal elongation of the true class distributions, so that there is some model mis-specification (i.e. the parametric model does not match the actual distributions) which is the common case in machine learning. The model parameters are the means and variances of Gaussian distributions.

The training data set consists of 200 samples per class, where only a few of them are labeled and the testing dataset consists of 200 samples per class. We ran experiments with 2, 4, and 6 labeled points, where we vary the hyper-parameter tuning the degree of importance of the unlabeled data (e.g. γ in IHA). To limit the bias of the choice of labeled training samples, all experiments are run 50 times with different random initial choices. In any case, the model parameters are initialized by setting the means of the isotropic Gaussians to the mean of the labeled samples, and by setting the variances to one.

Fig. 2 shows three examples of how the accuracy of the generative and the discriminative systems (we use the *MaxOne* variant of our approach) evolve with the iteration number. In the three experiments the models are trained with 4 labeled points for 50 iterations. These figures show typical behaviors of the method. Although IHA outperforms supervised learning in the three cases, the performance might be unstable and is not always improved every iteration. For instance, in Fig. 2-*a* the method starts with a satisfying initial performance, but the performance drops with the number of iterations. In Fig. 2-*b* the behavior is more chaotic but the final performance is again better than supervised learning. Less chaotic behavior is shown in Fig. 2-*c*, where after several small fluctuations the final performance is better than that of supervised learning.

Next, we compare the performances of IHA, HM and EM by setting the hyperparameter to their best values. Table 1 reports the percentage of runs in which one method outperforms the other. Note that the numbers do not always sum up to 100% as in some cases the same performance is achieved by both methods. One sees that HM performs better with two labeled points per class, that IHA and HM achieves similar performances with four labeled points, and that IHA performs better with six labeled points. In particular, it shows that IHA performs significantly better with six labeled points, where it outperforms HM in 72% of cases and Entropy Minimization in 96% of cases.

These results suggest than our method maybe requires that both generative and discriminative classifiers work well enough for the approach to work well (which is a necessity for co-training to work in practice too).

6. Experiments on real datasets

We describe in this section experimental results gained on financial time series and on handwriting data.

6.1. Datasets and settings

In this section, we first detail our datasets and we introduce the benchmark methods that we compared with our approach. Then, we present our experimental settings.

6.1.1. Datasets

The financial time series dataset consists of chart patterns. A chart pattern is a particular shape that occurs in a stock-exchange series (a series of the daily value of the stock index) and that has some predictive power for financial operators (see Fig. 3). We used two datasets of chart patterns, the first one (*CP4*) includes 448 series corresponding to the 4 most popular patterns *Head and Shoulders*, *Double Top*, *Reverse Head and Shoulders* and *Reverse Double Top*. The second dataset *CP8* includes 896 patterns from 8 classes, the four previous ones and four additional chart patterns: *Triple Top*, *Ascending Triangle* (and the reverse patterns). Sequences are first normalized and a feature vector is computed for each time (i.e. each day) based on the local shape of the series (we compute the slope, the height...). Both datasets are divided into 2 parts: a training dataset with 70 samples per class and a test set with 20 samples per class each.

The handwriting dataset is a subset of the benchmark IAM database (Marti and Bunke, 2002), which consists of images of handwritten letters extracted from English word images. Each image is transformed into a series of feature vectors by using a sliding window moving from the left to the right of the image and by computing a feature vector for every window position (Marti and Bunke, 2002). We used two versions of the dataset. A small dataset called *Small IAM* includes 23 classes (lowercase characters) and is divided into a training set with 200 samples per class and a test set with 50 samples per class. A bigger dataset called *Big IAM* includes 20 classes only (less represented classes have been removed) and consists of 2 600 samples per class in the training set and 600 samples per class in the test set.

In all experiments below, unless otherwise stated, we use 50 unlabeled training samples per class for experiments on *CP4* and on *CP8*, 150 unlabeled training samples per class for experiments on Small IAM and 500 unlabeled training samples per class for experiments on Big IAM.

6.1.2. Benchmark methods

We compared our approach with supervised training (HMMs and HCRFs initialized by HMMs, named *Supervised HCRF init HMM* hereafter) and with state of the art semi-supervised approaches for training HMMs and HCRFs. To this end, we extended to HCRFs two main semi-supervised approaches that have been used for CRFs, the first method is entropy minimization (Jiao, 2006) (we will refer to this method as *SSL entropy HCRF*), the second one has been proposed in Sokolovska (2011) (we will refer to this method as *SSL weighted HCRF*). We also compared our approach to the general co-training algorithm proposed in Wang and Zhou (2007) for learning two systems (HMMs and HCRFs) that operate on the same view. Finally, we will compare with a simple learning that consists in first initializing a HCRF system with a HMM system that has been trained (to maximize likelihood) in a semi supervised setting, second in relearning the HCRF in a supervised mode by regularizing around the HMM solution (we will refer to this method as *SSL HCRF init HMM*), this method is equivalent to the first iteration of IHA.

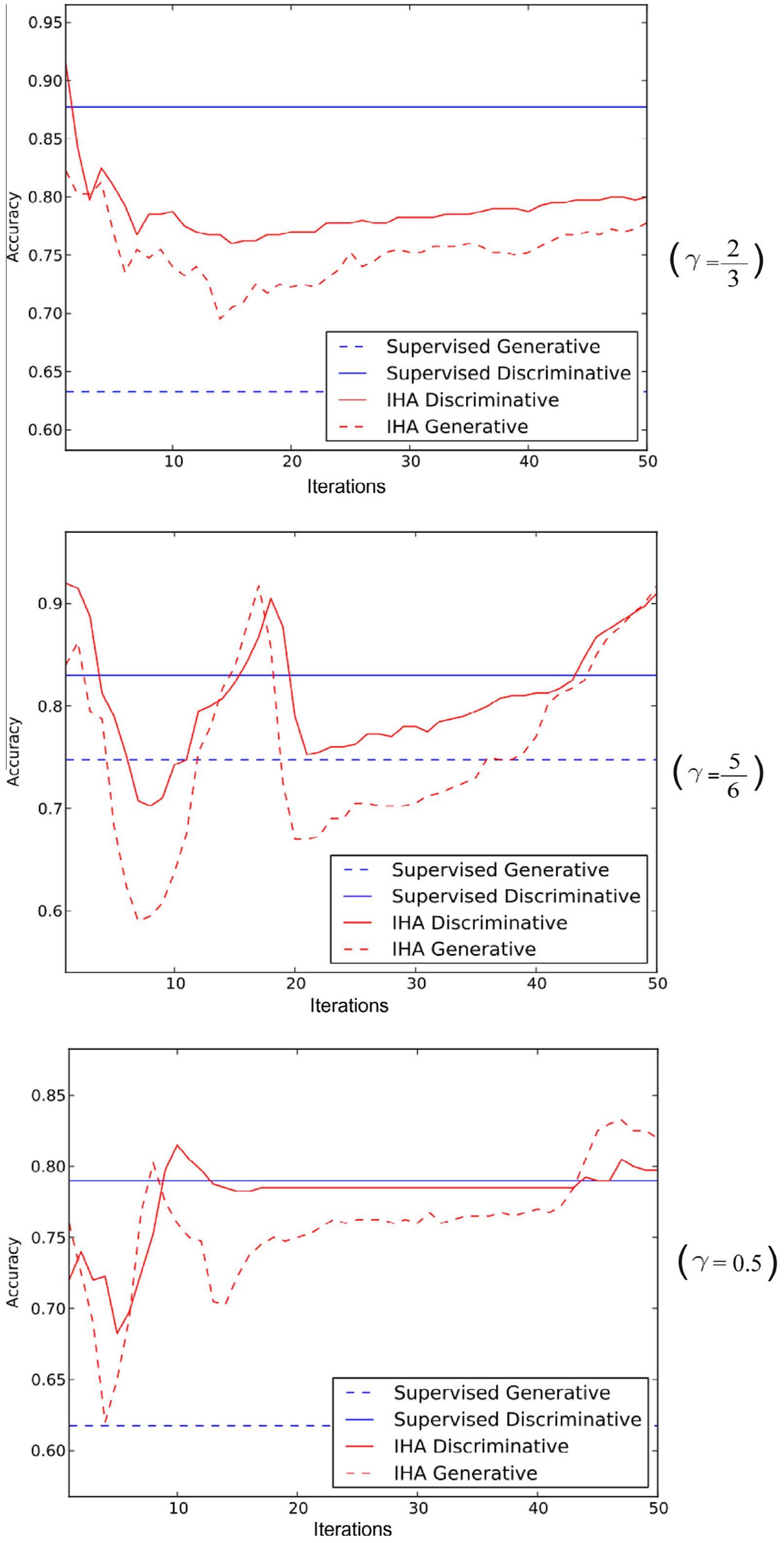

Fig. 2. Performances of the Iterative Hybrid Algorithm and of supervised training as a function of iteration number, as observed in three different runs for different values of γ. All experiments are performed on data with four labeled points.

6.1.3. Experimental settings

In experiments on *CP4*and *CP8*, we used one left–right HMM and HCRF per class which have either 4 or 6 states depending on the shape of the figures (see Fig. 3, e.g. the model of *Head and Shoulders* has six states since it is naturally composed of six segments). On the IAM dataset, we used left–right HMM and HCRF models with 8 states.

In any case, HMMs have a single Gaussian distribution with full covariance matrix as emission probability density. HCRF training is systematically performed through two steps: initialization by a

Table 1
Percentage of runs where the method on the left performs better than the method on the top, for 2, 4 and 6 labeled points per class.

	2 Samples per class			4 Samples per class			6 Samples per class		
	IHA (%)	HM (%)	EM (%)	IHA (%)	HM (%)	EM (%)	IHA (%)	HM (%)	EM (%)
IHA	0	16	36	0	40	72	0	72	96
HM	84	0	68	52	0	82	22	0	72
EM	64	24	0	28	14	0	2	16	0

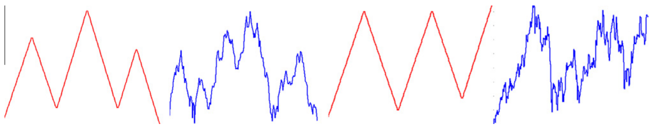

Fig. 3. From left to right: ideal shape of a Head and Shoulder pattern (HS), example of actual HS in the Dow Jones series, ideal shape of an Ascending Triangle pattern (AT), and example of an actually observed AT in the Dow Jones.

HMM system that has been trained through Maximum Likelihood Estimation followed by a retraining by optimization of the HCRF criterion (conditional likelihood) using stochastic gradient descent.

To limit the bias of choosing which training samples are labeled, we ran multiple times all experiments where we randomly choose labeled training samples and we report averaged results. In the following we provide first preliminary results gained with 4 runs while in the final results (Section 6.2.2), we performed 20 runs on IAM datasets and 60 runs on chart pattern datasets.

Note that it is not common to use a validation set in a semi-supervised learning because labeled samples are very few and more useful in the training set. In our experiments we perform training of each model (HMM, HCRF, etc.) through a fixed number of iterations, either 4 or 30, as specified in the text, and did not make use of any validation dataset.

6.2. Results

We first investigate the behavior of our approach and of its variants. Then, we compare in deep the behavior of the IHA approach and to those of benchmark methods.

6.2.1. Preliminary results

6.2.1.1. Comparison of variants of IHA. We compare first the behavior of the variants of our approach as detailed in Section 4.4. Table 2 reports accuracies for experiments with 5 labeled training samples per class for three datasets, CP4, CP8 and Small IAM. Supervised learning and initial semi-supervised learning of HMMs in IHA is performed with either 4 or 30 iterations. In every following loop of IHA, the retraining of the models (either HMMs or HCRFs) is performed with 4 iterations.

Table 2 shows that the *MaxProb* and the *WeightOne* variants are often close and provide the best results while the *SelectProb* and the *AllClasses* strategies are less efficient, especially the *AllClasses* strategy which sometimes degrade the supervised case. Actually the *AllClasses* strategy is very close to the *mixture*semi-supervised framework for HMMs, so that these results confirm those of Nigam et al. (2000); Inoue and Ueda (2003) where this method has been shown to eventually degrade supervised learning performance. One notes also that in most cases, running 30 training iterations degrades performances: the labeled training dataset is probably too small so that models tend to overfit. At the end, the *MaxProb* strategy significantly outperforms the purely supervised training, for both HMM and HCRFs, and appears as the best method among all variants for 4 training iterations. We only focus on this variant in the next experiments.

6.2.1.2. Influence of the number of labeled samples. Table 3 studies the influence of the number of labeled training samples per class on the performance of the supervised training and on IHA. We used from 1 to 10 labeled samples per class. Whatever the dataset, CP8 and Small IAM, HCRFs outperform corresponding HMMs. Also our

Table 2
Performances on the test set of supervised learning of HMMs and HCRFs (which is initialized by HMMs) compared to a few variants of our semi-supervised approach.

Database	Iterations	Supervised		AllClasses		MaxProb		WeightOne		SelectProb	
		HMM (%)	HCRF (%)	HMM (%)	HCRF (%)	HMM (%)	HCRF (%)	HMM (%)	HCRF (%)	HMM (%)	HCRF (%)
CP4	4	77.2	79.1	78.4	79.4	**85.3**	84.4	85.0	84.1	78.8	80.0
	30	77.5	78.8	78.4	79.1	**85.0**	84.7	84.4	84.1	80.9	80.6
CP8	4	62.0	61.1	61.6	61.6	63.4	**64.2**	62.7	64.1	62.7	64.1
	30	62.5	63.4	62.7	63.4	**66.9**	**66.9**	66.1	65.9	64.2	63.4
small IAM	4	36.9	38.9	37.8	39.0	40.4	41.5	40.2	**41.6**	37.9	39.3
	30	37.2	38.7	37.1	38.4	39.4	**40.1**	39.1	39.3	38.2	38.4

Table 3
Comparison on the test set of supervised training for HMMs and HCRFs initialized by HMMs to semi-supervised training with our approach (*MaxProb* variant) as a function of the number of labeled samples per class.

Labeled data	CP8				Small IAM			
	Supervised		IHA		Supervised		IHA	
	HMM (%)	HCRF (%)	HMM (%)	HCRF (%)	HMM (%)	HCRF (%)	HMM (%)	HCRF (%)
1	32.5	38	48.9	49.4	14.7	19.1	23.7	23.9
2	51.4	51.4	55.9	56.7	24.6	28.5	30.0	30.7
5	62.0	61.1	63.4	64.2	36.9	38.9	40.4	41.5
10	62.7	63.9	66.3	66.6	46.3	47.1	47.1	48.1

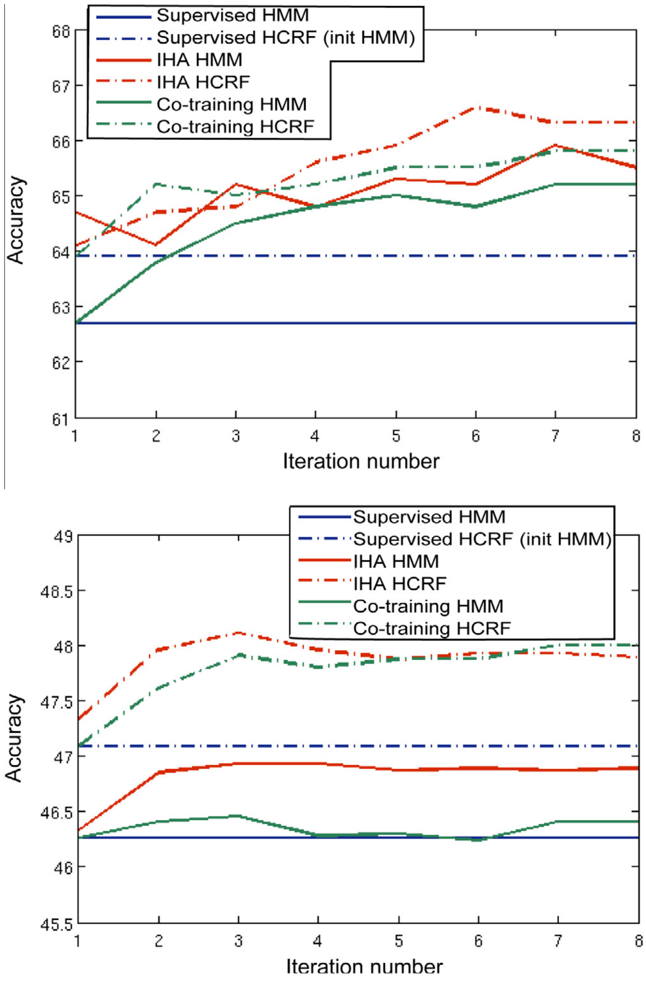

Fig. 4. Performance on the test set of HMMs and HCRFs models as a function of iteration number in IHA on *CP8*(top) and on *small IAM*(bottom).

approach systematically and significantly improves over supervised learning, both for learning HMMs and for learning HCRFs.

6.2.1.3. Evolution of the performance with iteration number in IHA. It is interesting to look at the evolution of the performance as a function of the iteration number in iterative semi-supervised algorithms.[3] Fig. 4 plots an example of typical curves for iterative algorithms such as co-training and IHA. These curves have been obtained with 10 labeled samples per class for CP8 and Small IAM datasets. We plot the performance of supervised learning, both for HMMs and for HCRFs, for comparison.

One sees that both iterative algorithms allow improving over supervised training. Note here that the performance of both HMMs and HCRFs increase almost monotonously until it converges. Note also that our approach may reach its best results after a few iterations (Small IAM dataset) or may require more iterations (CP8 dataset) to converge to an accurate solution. This depends on the datasets.

6.2.1.4. Influence of the number of unlabeled samples. Finally, we investigated the influence of the number of unlabeled data on the performance of our approach. Fig. 5 plots the accuracy as a function of the number of unlabeled samples while the number of labeled samples remains fixed. We used 5 labeled samples per class and from 25 to 500 unlabeled samples per class (we used the largest dataset for this experiment, *Big IAM*). We compare supervised learning, simple semi-supervised strategies, i.e. a standard strategy for semi-supervised learning of HMMs (mixture approach) and a simple strategy for learning HCRFs[4], and IHA (note that the simple semi-supervised learning for learning HCRFs corresponds to the first iteration of IHA).

The first point to note is that simple semi-supervised learning most often significantly outperforms supervised learning for both HMMs and HCRFs but accuracy increases up to a plateau where it fails taking more benefit from unlabeled data. The performances of both HMMs and HCRFs, learned with IHA increase steadily with the number of unlabeled data and allows learning even more accurate classifiers.

6.2.2. Comparison with state of the art semi-supervised methods

Finally, we compared more extensively our method with main state of the art semi-supervised methods on our four datasets (CP4, CP8, small IAM and Big IAM). All models are trained on 5 labeled samples per class and we use 50 unlabeled samples per class on the Chart Pattern datasets and 150 and 500 unlabeled samples per class on the Small IAM and Big IAM datasets. We report here averaged results gained with 20 runs on the IAM corpus, and with 60 runs on *CP4*and *CP8*. We provide the 95% confidence interval on all results. The 95% confidence intervals are specified by the width x of the interval around the reported mean accuracy m, it is indicated as $m\% \pm x$.

Note that since standard semi-supervised training of HMM do not always improve over supervised learning, we investigated our variants for learning HMMs in a semi-supervised way and report results of the best strategy (we will refer to it as *SSL HMM*). We use the *MaxProb* strategy for Chart Patterns and the *SelectProb* for IAM datasets. Also, as said previously we use the *MaxProb* strategy for IHA.

This table calls for a few comments. First, SSL learning systematically outperform supervised learning, for both HMM and HCRF systems. Second, our extension to HCRF learning of state of the art semi-supervised learning, i.e. the SSL entropy HCRF and the SSL weighted HCRF, behave very closely whatever the dataset, but yield only small improvement, if any, over simple SSL training. Third the co-training algorithm performs sometimes better and sometimes worse than these simple SSL methods for learning HCRFs. It appears to be less robust and maybe more difficult to tune. Finally, IHA most often outperforms all other methods for learning HCRF systems (see Table 4).

Considering HMM results now, it appears that IHA allows to learn HMM systems that significantly outperform supervised learning, standard HMM semi-supervised learning and also the co-training algorithm.

Although the confidence intervals are not small, the same trends may be observed whatever the dataset, which enforces our conclusions. Note also that improvements are bigger on the IAM datasets (which are more difficult tasks) than on the Chart Pattern datasets.

As a conclusion, our approach allows making use of unlabeled data to improve the behavior of both generative systems and discriminative ones. It compares well to state of the art methods for SSL learning and most often outperforms these. Importantly, a by-product of the algorithm is an efficient SSL trained generative system which significantly outperforms other SSL learning using these models.

[3] Note that one iteration stands here for a retraining of both the HMM system and the HCRF system.

[4] a HMM system learned with a mixture strategy and a HCRF system learned in a supervised setting from this HMM solution (as discussed in Section 3)

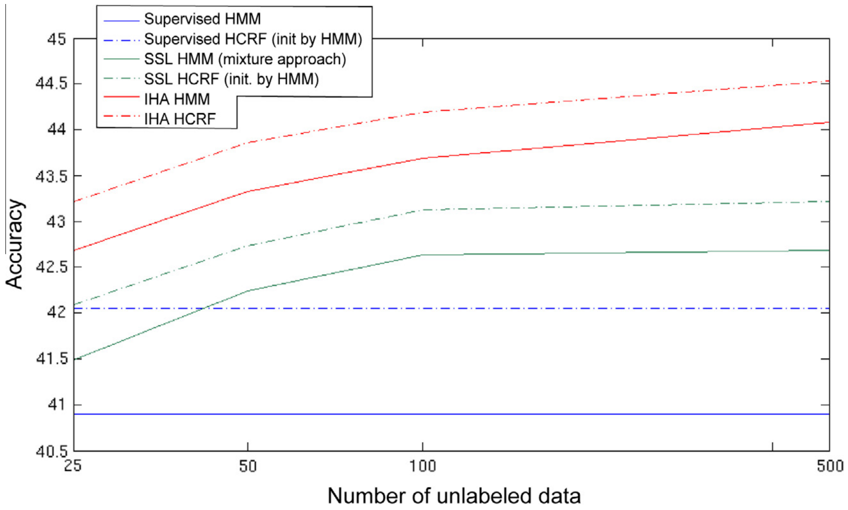

Fig. 5. Comparison of the accuracy of HMM and HCRF systems trained in a supervised and in a semi-supervised setting with HMM and HCRF systems learned with our iterative approach on the Big IAM dataset. Performance is plotted as a function of the number of unlabeled samples used.

Table 4
Comparison of our proposed semi-supervised HCRF and iterative framework with state of the art methods: Semi-supervised learning of HCRFs using entropy or weighted approaches and the general co-training algorithm (using one HMM system and one HCRF system initialized by others HMMs).

Method	CP4	CP8	Small IAM	Big IAM
Supervised HMM	78.5 ± 1.1	59.3 ± 0.9	35.8 ± 1.0	40.9 ± 0.9
Supervised HCRF init HMM	78.7 ± 1.1	59.7 ± 0.9	37.6 ± 1.0	42.0 ± 0.9
SSL HMM	83.8 ± 0.6	61.8 ± 0.9	36.6 ± 1.2	42.7 ± 1.0
SSL HCRF init HMM	83.9 ± 0.6	62.0 ± 1.00	37.6 ± 1.2	43.2 ± 1.0
SSL entropy HCRF	84.0 ± 0.6	62.0 ± 0.9	37.6 ± 0.9	43.2 ± 1.0
SSL weighted HCRF	83.9 ± 0.5	62.0 ± 0.9	37.7 ± 0.9	43.2 ± 1.0
Co-training HMM	83.5 ± 0.6	61.5 ± 0.9	35.7 ± 0.9	40.9 ± 0.9
Co-training HCRF	83.5 ± 0.7	61.9 ± 0.9	**39.5** ± 0.9	43.6 ± 0.9
IHA HMM	84.0 ± 0.5	62.1 ± 0.9	38.8 ± 1.0	44.1 ± 0.9
IHA HCRF	**84.2** ± 0.5	**62.4** ± 0.9	38.9 ± 1.0	**44.5** ± 0.9

7. Conclusion

We presented a framework for semi-supervised learning of a pair of generative and discriminative models. We investigated its behavior for learning a HMM system and a HCRF system for sequence classification. Our experimental results on artificial data and on two real datasets show that our strategy efficiently allows taking into account unlabeled data both for learning the discriminative models (HCRF) and the generative models (HMMs). It compares well to state of the art semi-supervised approaches that we investigated for learning HCRF in a semi-supervised setting and also to the well known co-training algorithm.

References

Baluja, S., 1998. Probabilistic modeling for face orientation discrimination: learning from labeled and unlabeled data. Advances in Neural Information Processing Systems 11, 854–860.

Bishop, C. M., Lasserre, J., 2007. Generative or discriminative? getting the best of both worlds. Bayesian Statistics 8, 3–24.

Blum, A., Mitchell, T., 1998. Combining labeled and unlabeled data with co-training. Morgan Kaufmann Publishers, pp. 92–100.

Bouchard, G., 2007. Bias-variance tradeoff in hybrid generative-discriminative models. In: ICMLA '07: Proceedings of the Sixth International Conference on Machine Learning and Applications, IEEE Computer Society, Washington, DC, USA, pp. 124–129.

Bouchard, G., Triggs, B., 2004. The trade-off between generative and discriminative classifiers. In: Proceedings in Computational Statistics, 16th Symposium of IASC, pp. 721–728.

Collins, M., 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In: EMNLP.

Cozman, F.G., Cohen, I., 2002. Unlabeled data can degrade classification performance of generative classifiers. In: Fifteenth International Florida Artificial Intelligence Society Conference, pp. 327–331.

Do, T.-M.-T., Artières, T., 2005. Conditional random field for tracking user behavior based on his eye's movements. In: NIPS'05 Workshop on Machine Learning for Implicit Feedback and User Modeling, Whistler, BC, Canada.

Do, T.-M.-T., Artières, T., 2006. Conditional random fields for online handwriting recognition. In: Guy Lorette (Ed.), Tenth International Workshop on Frontiers in Handwriting Recognition, Université de Rennes 1, Suvisoft, La Baule, France.

Do, T.-M.-T., Artières, T., 2009. Large margin training for hidden Markov models with partially observed states. In: Bottou, L., Littman, M. (Eds.), Proceedings of the 26th International Conference on Machine Learning. Omnipress, Montreal, pp. 265–272.

Frinken, V., Peter, T., Fischer, A., Bunke, H., Do, T.-M.-T., Artieres, T., 2009. Improved handwriting recognition by combining two forms of hidden markov models and a recurrent neural network. In: CAIP '09: Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns. Springer-Verlag, Berlin, Heidelberg, pp. 189–196.

Grandvalet, Y., Bengio, Y., 2005. Semi-supervised learning by entropy minimization. Network 17 (5), 529–536.

Gunawardana, A., Mahajan, M., Acero, A., Platt, J.C., 2005. Hidden conditional random fields for phone classification. In: Interspeech, pp. 1117–1120.

Haffari, G., Sarkar, A., 2008. Homotopy-based semi-supervised hidden markov models for sequence labeling. In: COLING, pp. 305–312.

Inoue, M., Ueda, N., 2003. Exploitation of unlabeled sequences in hidden markov models. IEEE Transactions On Pattern Analysis and Machine Intelligence 25, 1570–1581.

Jiao, F. , 2006. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In: COLING/ACL, COLING/ACL, pp. 209–216.

Ji, S., Watson, L.T., Carin, L., 2009. Semisupervised learning of hidden markov models via a homotopy method. IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2), 275–287.

Juang, B., Katagiri, S., 1992. Discriminative learning for minimum error classification. In: IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 40 (12).

Juang, B.H., Rabiner, L.R., 1990. The segmental K-means algorithm for estimating parameters of hidden Markov models. In: IEEE Transactions on Acoustics, Speech and Signal Processing.

Khine, S.Z.K., Nwe, T.L., Li, H., 2008. Singing voice detection in pop songs using co-training algorithm. In: ICASSP, IEEE, pp. 1629–1632.

Lafferty, J.D., McCallum, A., Pereira, F.C.N., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 282–289.

Lasserre, J., 2008. Hybrid of generative and discriminative methods for machine learning (thesis).

Mann, G.S., McCallum, A., 2010. Generalized expectation criteria for semi-supervised learning with weakly labeled data. Journal of Machine Learning Research 11, 955–984.

Marti, U., Bunke, H., 2002. A full english sentence database for off-line handwriting recognition. In: ICDAR.

Mérialdo, B., 1994. Tagging english text with a probabilistic model. Computational Linguistics 20 (2), 155–171.

Minka, T.P., 2005. Discriminative Models, Not Discriminative Training. Microsoft Research, Cambridge.

Morency, L.-P., Quattoni, A., Darrell, T., 2007. Latent-dynamic discriminative models for continuous gesture recognition. In: CVPR.

Morgan, N., Bourlard, H., 1995. An introduction to the hybrid HMM/connectionist approach. In: IEEE Signal Processing magazine.

Ng, A.Y., Jordan, M.I., 2001. On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes. In: NIPS, pp. 841–848.

Nigam, K., McCallum, A., Thrun, S., Mitchell, T.M., 2000. Text classification from labeled and unlabeled documents using EM. Machine Learning 39 (2/3), 103–134.

Quattoni, A., Wang, S.B., Morency, L.-P., Collins, M., Darrell, T., 2007. Hidden-state conditional random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (10), 1848–1852.

Rabiner, L.R., 1989. A tutorial on hidden markov models and selected applications in speech recognition. In: Proceedings of the IEEE, pp. 257–286.

Reiter, S., Schuller, B., Rigoll, G., 2002. Hidden conditional random fields for meeting segmentation, In: ICME, pp. 639–642.

Sha, F., Saul, L.K., 2007. Large margin hidden markov models for automatic speech recognition. In: Scholkopf, B., Platt, J., Hoffman, T. (Eds.), Advances in Neural Information Processing Systems, 19. MIT Press, pp. 1249–1256.

Sokolovska, N., 2011. Aspects of semi-supervised and active learning in conditional random fields. In: ECML/PKDD (3), pp. 273–288.

Soullard, Y., Artieres, T., 2011. Hybrid hmm and hcrf model for sequence classification. In: Proceedings of the European Symposium on Artificial Neural Networks (ESANN 2011), Computational Intelligence and Machine Learning, Bruges, Belgium, pp. 453–458.

Soullard, Y., Artieres, T., 2011. Iterative refinement of hmm and hcrf for sequence classification. In: IAPR Workshop on Partially Supervised Learning (PSL).

Suzuki, J., Fujino, A., Isozaki, H., 2007. Semi-supervised structured output learning based on a hybrid generative and discriminative approach. In: EMNLP-CoNLL, pp. 791–800.

Vinel, A., Do, T.M.T., Artières, T., 2011. Joint optimization of hidden conditional random fields and non linear feature extraction. In: ICDAR, pp. 513–517.

Wang, W., Hua Zhou, Z., 2007. Analyzing co-training style algorithms. In: Proceedings of the 18th European Conference on Machine Learning.

Wang, J., Shen, X., Pan, W., 2009. On efficient large margin semisupervised learning: method and theory. Journal of Machine Learning Research 10, 719–742.

Wang, Y., Haffari, G., Wang, S., Mori, G., 2009. A rate distortion approach for semi-supervised conditional random fields. In: NIPS, pp. 2008–2016.

Woodland, P., Povey, D., 2002. Large scale discriminative training of hidden markov models for speech recognition, Computer Speech and Language.

Zhu, X., Goldberg, A.B., 2009. Introduction to Semi-Supervised Learning. In: Introduction to Semi-Supervised Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.