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
firstname.lastname@lip6.fr

Abstract

Although semi-supervised learning has generated great interest for designing classifiers on vector data there has been comparatively very few works on semi-supervised learning for structured outputs like sequences. We investigate here semi-supervised approaches for learning hidden state conditional random fields that have been proposed recently for sequence and signals classification. In particular we propose a new approach to deal with this problem that relies on an iterative joint learning of a pair of discriminative-generative models, namely Hidden Markov Models (HMMs) and Hidden Conditional Random Fields (HCRFs). The method builds on rather simple strategies for semi-supervised learning of HMMs and on strategies for learning a HCRF from a HMM. This algorithm has connections with few previous works such as co-training. We investigate the behavior of the method on artificial data and provide experimental results on two real datasets for handwriting letters classification and chart pattern recognition. All along the experimental section, we compare our approach with state of the art semi-supervised methods, entropy minimization and co-training.

Keywords: Hidden Markov Models, Hidden Conditional Random Fields, semi-supervised learning, co-training

1. Introduction

Sequence classification and sequence labeling are fundamental tasks occurring in many application domains such as speech, financial time series, and handwriting. The most popular method for such tasks are the well known Hidden Markov Models (HMMs) [1]. HMMs are generative models which are trained to maximize the joint likelihood of observation sequences and of their labeling. HMMs benefit from efficient algorithms both for inference and for training but suffer few severe drawbacks. In particular they are traditionally learned via maximum likelihood estimation, which is a non discriminative training criterion.

Although some attempts have been made to overcome this limitation by learning discriminatively HMM systems through the optimization of a discriminant criterion like minimum error classification [2], perceptron loss [3], maximum mutual information [4], margin maximization [5, 6]. A more straightforward way to reach higher discriminative power is to define a model of the posterior conditional probability (i.e. the probability of the label given the input sequence). Hidden Conditional Random Fields (HCRFs) are such models. They are a variant of Conditional Random Fields (CRFs) [7] that make use of hidden states to account for the underlying structure of the data (alike HMMs). They have been used for various signal labeling tasks, in particular for speech signals [8], [9], eye movements [10], handwriting [11, 12], gestures and images [13], financial time series [14].

When building a classification system, whatever the model one chooses to design a sequence labeling system one has first to gather, then to label, a sufficiently large training corpus. This always comes with a cost that may make problematic the design of a good system. This motivated the study of semi-supervised learning (SSL), which has been proposed first for vector data. SSL aims at learning classifiers based on both labeled samples (usually few) and unlabeled samples (usually many). A number of SSL methods have been proposed, like entropy based methods [15], margin based methods [16], co-training algorithms [17] (see [18] for reviews).

Few works have investigated semi-supervised learning for sequential and more generally structured data. Some of these investigated HMMs semi-supervised learning for speech recognition and text classification tasks [19], [20],[21], but the conclusion of these works are rather mitigated, where SSL has been shown to eventually degrade performance with respect to purely supervised training [22], [23]. It appears in the literature that SSL may be less efficient for learning more complex models, such as HMMs which include hidden states to deal with partially observed data. Besides, some other works attempted to learn CRFs in a semi-supervised setting for language processing and biological problems, yielding some significant improvements [24], [25]. It is worth noticing that few of these works rely on designing a hybrid model mixing HMMs and CRFs where HMMs are learned in a semi-supervised way, making indirectly the learning of CRFs to be semi-supervised [25]. At the end we are not aware of any work today on SSL algorithms for complex discriminative models such as HCRFs excepted in our previous works [26].

We propose in this study to investigate the relevance of previous SSL

schemas to the learning of HCRF and we propose a new algorithm for semi-supervised learning of HCRFs. It relies on an iterative joint learning of a pair of generative and discriminative models, namely HMMs and HCRFs. It relies on the relative easiness of SSL for HMMs and on recent results showing how one can initialize a HCRF from a HMM system [8]. In our approach, HMMs learning makes explicit use of both labeled and unlabeled data while HCRF learning is purely supervised and indirectly exploits unlabeled data through its initialization from the HMM system. All along the paper we focus on sequence classification where one wants to assign a single label to an input sequence but extension to sequence labeling is in our mind and our next step.

We first recall basics of HMMs and HCRFs in section 2 and we discuss related works on semi-supervised learning in section 3. Then, we detail our approach in section 4. We report experimental results on artificial data in section 5 and investigate the behavior of our approach on handwriting recognition and on financial chart pattern classification in section 6.

2. Markovian models for sequence classification

In this study, we focus on sequence classification where training samples are couples $(\mathbf{x}, y)$, where we note $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_T) \in \mathcal{X}$ an input sequence of length T (sequences are noted in bold), with each frame $\mathbf{x}_t \in \mathbb{R}^d$ being a d-dimensional feature vector which characterizes locally the input sequence, and $y \in \mathcal{Y}$ is its class (i.e. label).

2.1. From Hidden Markov Models to Hidden Conditional Random Fields

We briefly recall basics of Hidden Markov Models (HMMs) then we detail Hidden Conditional Random Fields (HCRFs). Note that we focus on sequence classification all along the presentation.

HMM are generative models which define a joint probability over an observation sequence and its class $p(\mathbf{x}, y)$. To account for variability of observations HMMs include hidden states which are not observed. Hence training data are incomplete in that the state sequence corresponding to a particular observation sequence is unknown. As a result the joint probability of an observed sequence and of its class is the sum over all hidden state sequences of the joint probability of the observed sequence and of the hidden state sequence:

$$p(\mathbf{x}, y|\Theta) = \sum_{\mathbf{h}} p(\mathbf{x}, \mathbf{h}|y, \Theta)\, p(y|\Theta) \tag{1}$$

where Θ stands for the HMM parameters $\mathbf{h}$ denotes a hidden state sequence, with $\forall t, h_t \in S$ with $S = \{s_1, ..., s_Q\}$ being the set of the Q states of the model. In the formula above the summation over $\mathbf{h}$ is taken over all $\mathbf{h}$ that match with the labeling y, that we will note $S(y)$ in the following.

The joint probability $p(\mathbf{x}, \mathbf{h}|y)$ may be factorized according to (taking the logarithm):

$$\log p(\mathbf{x}, \mathbf{h}|y) \quad = \log p(h_1) + \textstyle\sum_{t=2}^{T} \log p(h_t|h_{t-1}) + \sum_{t=1}^{T} \log p(\mathbf{x}_t|h_t) \tag{2}$$

where, using HMM's standard terminology, $p(h_1)$ are initial probabilities, $p(h_t|h_{t-1})$ are transition probabilities, and $p(\mathbf{x}_t|h_t)$ are emission probabilities.

HMMs are usually learned in a non discriminative way to maximize the likelihood of training data. Traditionally for dealing with signal like data (speech, handwriting, etc), one uses a left-right HMM topology for every class, where transitions are allowed from any state to itself or to the next state.

Hidden CRFs (HCRFs) have been proposed as an extension of CRFs (that were initially proposed in [7] for text data) for dealing with more complex and structured data [27] [11] [8]. In CRF-based systems there is usually one state per class (e.g. a POS tag) while there are several states corresponding to a given class in HCRF, alike in HMMs. Given a HCRF with parameters Λ the class conditional probability of a class y given an input sequence $\mathbf{x}$ defined as:

$$p(y|\mathbf{x},\Lambda) \quad = \frac{1}{Z(\mathbf{x},\Lambda)} \sum_{\mathbf{h}\in S(y)} exp^{<\Lambda,\Phi(\mathbf{x},y,\mathbf{h})>} \tag{3}$$

where $\Phi(\mathbf{x},y,\mathbf{h})$ is a joint feature vector corresponding to a state sequence $\mathbf{h}$, and $Z(\mathbf{x},\Lambda) = \sum_{y'} \sum_{\mathbf{h'}\in S(y')} exp^{<\Lambda,\Phi(\mathbf{x},y',\mathbf{h'})>}$ is a normalization term. When given an input sequence $\mathbf{x}$, its predicted class is determined according to $argmax_y p(y|\mathbf{x},\Lambda)$.

To make the model tractable one assumes the feature vector Φ to be decomposable. Usually this decomposition relies on a Markov network encoding conditional dependencies between random variables to infer (y and $\mathbf{h}$). In sequence modeling one often considers a decomposition over time steps :

$$\Phi(\mathbf{x},y,\mathbf{h}) = \sum_t \phi(\mathbf{x},y,\mathbf{h},t) \tag{4}$$

where $\phi(\mathbf{x}, y, \mathbf{h}, t)$ is a *local* feature function. More precisely on exploits a Markov network with two types of cliques: transition cliques involving two successive states and local cliques involving a state at a particular time and the observation at that time. With these assumptions, the conditional probability above is usually rewritten as:

$$p(y|\mathbf{x}, \Lambda) = \frac{1}{Z(\mathbf{x}, \Lambda)} \sum_{\mathbf{h} \in S(y)} exp^{\sum_t <\boldsymbol{\lambda}^{trans}, \boldsymbol{\phi}^{trans}(\mathbf{x}, y, h_t, h_{t-1})> + <\boldsymbol{\lambda}^{loc}, \boldsymbol{\phi}^{loc}(\mathbf{x}, y, h_t)>} \quad (5)$$

where $\boldsymbol{\lambda}^{loc}$ is the subset of Λ weighting the features for each state, $\boldsymbol{\lambda}^{trans}$ is the subset of Λ weighting the transition features between states, $\boldsymbol{\phi}^{loc}$ and $\boldsymbol{\phi}^{trans}$ are joint feature vectors for transition and local cliques.

2.2. Learning

HMMs are traditionally learned via the Baum-Welsh algorithm [28] to maximize the likelihood of the training data:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{n} log(p(y^i, \mathbf{x}^i|\Theta)) \quad (6)$$

where Θ denotes the parameter set of the HMMs of all classes, and the superscript i refers to the number of the training sample, i.e. $\mathbf{x}^i$ stands for the i^{th} training sequence and y^i for its class.

Alike CRFs, HCRFs are usually learned to maximize the conditional likelihood:

$$\mathcal{L}(\Lambda) = \sum_{i=1}^{n} log(p(y^i|\mathbf{x}^i, \Lambda)) \quad (7)$$

The optimization is traditionally performed with gradient like algorithms, stochastic gradient [29], or second order extensions like LBFGS [30].

One limit of HCRFs lies in the non convexity of the training criterion, which comes from the introduction of hidden states just like for HMMs. This makes training sensitive to initialization and easily lead to overfitting. To overcome this problem few solutions have been proposed to initialize HCRFs. The most interesting one has been proposed in [8], it consists in learning first a HMM system, then to initialize the HCRF parameters so that it reproduces the classification behavior of the HMMs (note that the HCRF topology must match the HMM one). We briefly explain how this can be done.

The key point is that the joint log likelihood of an input sequence and of a sequence of states, as computed by a HMM, may be written as a dot product between a particular parameter vector and a joint feature map depending on the class, the sequence of hidden states and the input sequence. Indeed, noting Θ the parameter set of an HMM, for any state sequence $\mathbf{h}$ we have:

$$\begin{aligned}\log p(\mathbf{x}, y, \mathbf{h}|\Theta) = &\log(\pi_{h_1}) + \log(p(\mathbf{x}_1|h_1)) \\ &+ \sum_{t=2}^{T} \Big(\log p(h_t|h_{t-1}) + \log p(\mathbf{x}_t|h_t, \boldsymbol{\mu}_{h_t}, \Sigma_{h_t} \Big)\end{aligned}$$

with, following [31]:

$$\begin{aligned}\log p(\mathbf{x}_t|h_t, \Theta) = \frac{1}{2}\Big(&\mathbf{x}_t^T \Sigma_{h_t}^{-1} \mathbf{x}_t - \mathbf{x}_t^T \Sigma_{h_t}^{-1} \boldsymbol{\mu}_{h_t} - \boldsymbol{\mu}_{h_t}^T \Sigma_{h_t}^{-1} \mathbf{x}_t \\ &+ \boldsymbol{\mu}_{h_t}^T \Sigma_{h_t}^{-1} \boldsymbol{\mu}_{h_t} - \log((2\pi)^d |\Sigma_{h_t}|)\Big)\end{aligned}$$

Hence, we may write:

$$\log p(\mathbf{x}, y, \mathbf{h}|\Theta) =< \boldsymbol{\lambda}, \boldsymbol{\phi}(\mathbf{x}, y, \mathbf{h}) >$$

which may be rewritten, using appropriate definitions (see Appendix A for details), as:

$$< \boldsymbol{\lambda}, \boldsymbol{\phi}(\mathbf{x}, y, \mathbf{h}) >= \sum_t < \boldsymbol{\lambda}^{trans}, \boldsymbol{\phi}^{trans}(\mathbf{x}, y, h_t, h_{t-1}) > + < \boldsymbol{\lambda}^{loc}, \boldsymbol{\phi}^{loc}(\mathbf{x}, y, h_t) >$$

The above result yields an efficient learning procedure for learning HCRFs for sequence classification. First, one learns a HMM system with one HMM per class. Then one initializes a HCRF system with the same topology with the above formulas. This HCRF system outputs exactly the same classification decision as the HMM system. Finally one uses the standard discriminative conditional likelihood criterion of HCRFs to fine tune the HCRF system. At the end, the 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.

3. State of the art in semi-supervised learning

Although semi-supervised learning has received great attention in the last ten years there has been only few attempts for dealing with structured data and complex models like CRFs and HCRFs. Most works concern vector data, with some exceptions like [32]. We first review main categories of semi-supervised learning methods then we present in more details works that have focused on SSL for sequential data and in particular for markovian models.

3.1. Generic methods

We provide here a brief overview of the main categories of methods, a detailed survey may be found in [33]. Some of these methods are dedicated to generative models, some on discriminative models and finally few rely on a mix between generative and discriminative models.

Mixture methods relying on the learning of a mixture of generative models through Expectation Maximization have shown to improve performance by using unlabeled data, in the cases when the classes consist of well clustered data. For instance [19] apply the EM algorithm on mixture of multinomial distributions for the task of text classification. They showed that the resulting classifiers perform better than those trained only on labeled data. Besides [34] used the same algorithm on a face orientation discrimination task.

Co-training has been popularized by [17]. It assumes that the features (samples are feature vectors) can be split in two sets (views), each of which is sufficient to train a good classifier and they are conditionally independent of one another given the class. Initially, two separate classifiers are trained on the labeled data with each feature set. Then, each classifier classifies the unlabeled data and "teaches" the other classifier with the unlabeled examples it feels most confident about. Each classifier is re-trained with the new labels and the process is repeated. In real tasks, having two sufficient and redundant views as the standard framework [17] is uncommon. [35] extend this approach to two learners which are not necessary trained on two different views and prove that this may enhance the combination of the two models against individual systems. Thus, they show that the co-training process may work well with two classifiers working on the same view of the data provided the

models are enough different. Also, the co-training offers more guarantee to work well if the two classifiers are of comparable accuracy.

Graph-based semi-supervised methods define a graph where the nodes are labeled and unlabeled examples in the dataset, and edges reflect similarity of the examples. Different algorithms operating on the graph have been proposed, including [36, 37]. These methods achieve good performance when similar instances in the data set have similar labels.

Self-training is another commonly used technique. A classifier is first trained with the small amount of labeled data and then it is used to classify the unlabeled data. Then, the most confident unlabeled points, with their predicted labels, are added to the training set. The classier is re-trained and the procedure is repeated. This method has been successfully applied to several natural language processing tasks in [38, 39].

Entropy Minimization Method has been introduced in [15] by Grandvalet and Bengio. They use the label entropy on unlabeled data as a regularizer. By minimizing the entropy, the method assumes a prior which prefers minimal class overlap. For a training dataset of L labeled data $\{(\mathbf{x}^1, y^1), ..., (\mathbf{x}^L, y^L)\}$ and of U unlabeled data $\{\mathbf{x}^{L+1}, ..., \mathbf{x}^{L+U}\}$, they propose maximizing the following objective function:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{L} \log p(y^i|\mathbf{x}^i, \Theta) + \gamma \sum_{j=L+1}^{L+U} \sum_{y' \in \mathcal{Y}} p(y'|\mathbf{x}^j, \Theta) \log p(y'|\mathbf{x}^j, \Theta) \qquad (8)$$

where, γ is a hyper parameter of the algorithm controlling the influence of the unlabeled data.

Finally *hybrid methods* have been proposed that mix generative and discriminative models [40] and [41]. [41] proposed first a convex combination of

the objective functions of a generative and of a discriminative model while [40] proposed to optimize the following objective function :

$$\mathcal{L}(\Theta, \Lambda) = \sum_{i=1}^{L} \log p(y^i|\mathbf{x}^i, \Lambda) + \sum_{j=1}^{L+U} \log p(\mathbf{x}^j|\Theta) + log(p(\Theta, \Lambda)) \qquad (9)$$

where the prior $p(\Theta, \Lambda)$ allows blending generative and discriminative approaches. In the particular case where the prior is uniform the generative and discriminative models are decoupled so that the discriminative model is learned in a fully supervised setting. In the case where the prior is peaked on $\Theta = \Lambda$ the two models are constrained so that one recovers the approach in [41]. Finally if the prior is smoother, 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 setting.

3.2. Semi-supervised learning of markovian models for sequence classification

A number of works have proposed methods for semi-supervised learning of HMMs and of CRFs, they belong to above categories.

Mixture approach. The Mixture approach has been applied to HMMs in [19], [20]. In this setting, the likelihood-based criterion is defined as:

$$\mathcal{L}(\Theta) = \frac{(1-\gamma)}{L} \sum_{i=1}^{L} log\, p(\mathbf{x}^i, y^i|\Theta) + \frac{\gamma}{U} \sum_{j=L+1}^{L+U} log\, p(\mathbf{x}^j|\Theta) \qquad (10)$$

where the parameter $\gamma \in [0, 1]$ allows tuning the respective influence of labeled and of unlabeled data. The fully supervised and the fully unsupervised

cases are special cases when γ is respectively set to 0 and to 1 [42]. This approach has been observed to eventually degrade the performances for learning HMMs [22, 23].

Entropy minimization. [32] devises a semi-supervised variant of the support vector machine based on a co-training algorithm. The main approach extends the minimum entropy regularization framework [42] to CRFs with a regularized objective function that combines unlabeled conditional entropy and labeled conditional likelihood [24] :

$$\mathcal{L}_\gamma(\Lambda) = -\frac{||\Lambda||^2}{2} + \sum_{i=1}^{L} log\, p(y^i|\mathbf{x}^i, \Lambda) + \gamma \sum_{j=L+1}^{L+U} \sum_{y \in \mathcal{Y}} p(y|\mathbf{x}^j, \Lambda) log\, p(y|\mathbf{x}^j, \Lambda) \quad (11)$$

Similarly, [43], relying on information theory, defines the objective function as the combination of the conditional likelihood on labeled data and of the mutual information on unlabeled data .

Hybrid approach. few works have investigated combining a discriminative model trained on labeled data and a generative model trained with additional unlabeled samples. [44] suggested to discriminatively train a blending of I-units of discriminative models (CRFs) trained on labeled data and J-units of generative models (HMMs) whose learning exploits additional unlabeled data. In the same line of work, [25] proposed a weighted CRF where weights are marginal probabilities of input sequences obtained by a generative model trained in a semi-supervised manner :

$$\mathcal{L}(\Lambda) = \sum_{i=1}^{L} q(\mathbf{x}^i) \frac{1}{N_{\mathbf{x}^i}} log\, p(y^i|\mathbf{x}^i, \Lambda) \quad (12)$$

where $N_{\mathbf{x}}$ is the number of times $\mathbf{x}$ has been observed in the training set (data are discrete) and $q(\mathbf{x})$ is the marginal probability of observations which come from the generative model.

Co-training. Co-training has also been investigated with some success for generative markovian models. In particular, [45] applied the standard co-training algorithm with HMMs models for singing voice detection and [46] experimented co-training on HMMs and neural networks for handwriting recognition.

4. Joint semi-supervised learning for HMMs and HCRFs

Designing semi-supervised learning algorithms for HCRF has not really been studied up to now. A first solution is to extend traditional SSL approaches to HCRFs. In particular we implemented methods that have been proposed for CRFs (i.e HCRFs without hidden states) and we will investigate the behavior of the methods in the experiments section. Here we describe a new approach where we jointly learn iteratively a HMM and a HCRF system. We first present the motivation of this work then we present in details the method.

4.1. Motivation

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

On the one hand, generative approaches (e.g. HMMs) rely on the learning of one model per class and build a distribution over observations. These approaches may exhibit a higher bias and a lower variance than discriminative models [41]. On the other hand, the discriminative approach (e.g. HCRFs) directly model the conditional probability distribution which is more related to the classification goal. When the training set size increases towards infinity, these approach usually exhibits better asymptotic performance (i.e. accuracy) than generative models but the convergence to their optimal behavior may be slower. It may then happen that generative models reach their asymptotic performance faster than discriminative ones, i.e. with a smaller training set size [47]. From these general comments it may happen that generative models may be more accurate with a small training dataset while discriminative models are more accurate when the training set size increases. Hence, mixing the two to get the best of the two worlds in any situation makes sense and is definitely appealing.

Besides, it is worth noticing that generative models allow simple semi-supervised learning through the use of mixture models and of an EM learning scheme [19] while semi-supervised training in discriminative models is less straightforward, which is particularly true for markovian models with hidden states as we discussed in previous section.

Our goal is to take advantage of the strenghts of each approach. Since generative models may be easily learned in a semi-supervised setting and since they may be less sensitive to overfitting, a first reasonable idea is to learn first a HMM system in SSL setting, then to learn a discriminative HCRF system initialized from the HMM system using the strategy described in

section 2.2 and Appendix A. To prevent overfitting of the HCRF system, one can train it with a regularized likelihood criterion where the regularization term constrain the solution to stay close from the HMM solution.

Going further, one may consider that the HCRF system is more powerful from the HMM it was initialized from so that one can use it to label the unlabeled data for HMM retraining. And again, one can learn a new HCRF system from the new HMM system, etc. Hence our approach bares some similarity with co-training but borrows some idea to [48] as well.

4.2. Iterative Hybrid Algorithm (IHA)

Our learning method, which we call Iterative Hybrid Algorithm (IHA), is an iterative algorithm that blend generative and discriminative models in a semi-supervised framework. The main idea is to use the generative model to incorporate the additional information brought by the unlabeled data (U), and to use the discriminative model to achieve better classification accuracy. The overall algorithm is illustrated in Figure 1. Initially, we train a generative model on labeled and unlabeled data ($L \cup U$). In the main loop of the algorithm, we train a discriminative model on L, constraining its parameters to be close to the ones of the generative model it is initialized from. Then, we use the discriminative model to label part (or all) of U which we use with L to retrain a generative model in a supervised mode. We repeat this process a number of iterations or until convergence.

More formally, the method can be described as follows. As before the parameters of the discriminative models are denoted as Λ, and the parameters of the generative models are denoted as Θ.

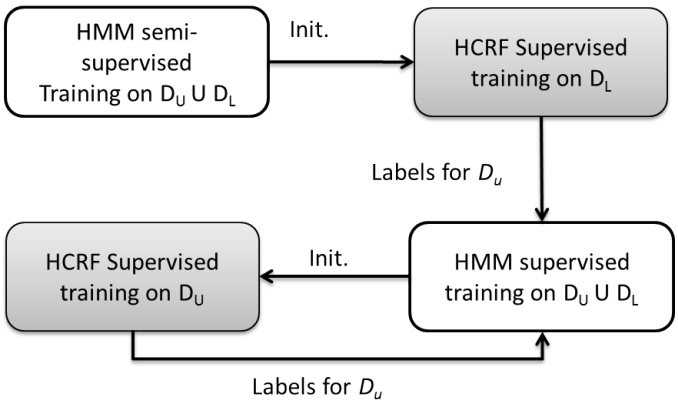


Figure 1: Semi-supervised strategy embedding HMM and HCRF learning, D_L and D_U denote the sets of labeled and unlabeled training sequences.

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

$$\Theta^{(1)} = argmax_{\Theta}\Big(\frac{\gamma}{L}\sum_{i=1}^{L}\log p(\mathbf{x}^i, y^i|\Theta) + \frac{(1-\gamma)}{U}\sum_{j=L+1}^{L+U}\log\sum_{y'\in\mathcal{Y}} p(\mathbf{x}^j, y'|\Theta)\Big) \tag{13}$$

2. Loop for a fixed number of iterations or until convergence. At iteration k:

   2.1. Supervised Learning of Λ on L yielding $\Lambda^{(k)}$, starting from $\Theta^{(k-1)}$:

$$\Lambda^{(k)} = argmax_{\Lambda}\sum_{i=1}^{L}\log p(y^i|\mathbf{x}^i, \Lambda) - \frac{1}{2}\|\Lambda - \Theta^{(k-1)}\|^2 \tag{14}$$

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

$$\forall i \in [L+1, L+U], \hat{y}^i = \arg\max_{y\in\mathcal{Y}} p(y|\mathbf{x}^i, \Lambda^{(k)}) \tag{15}$$

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

$$\Theta^{(k)} = argmax_{\Theta}\Big(\frac{\gamma}{L}\sum_{i=1}^{L}\log p(\mathbf{x}^i, y^i|\Theta) + \frac{(1-\gamma)}{|U|}\sum_{j=L+1}^{L+U_{Labeled}} p(\hat{y}^j|\mathbf{x}^j, \Theta)\log p(\mathbf{x}^j|\Theta)\Big) \quad (16)$$

The only hyper parameter in the model is γ, which controls the influence of the unlabeled data. Larger values of γ imply more influence of U. The final output of the method are the parameters of the discriminative models Λ (and the parameters of the generative models Θ) which are used to predict labels for new samples.

4.3. Alternative strategies

We investigated four variants of our method that differ by the way the generative system is retrained in step 2.3 of the algorithm described in previous section. To simplify notations, we note $U = U_{Labeled}$ and we define, for any class $c \in \mathcal{Y}$, L_c to be the number of labeled data in class c.

First we may use every unlabeled sample to retrain every class model. To do that an intuitive idea is to weight the contribution of an unlabeled sample by its posterior probabilities as given by the discriminative system. In this case, our algorithm comes close to the standard semi-supervised framework of generative models. The objective for learning the whole HMM system writes:

$$\mathcal{L}(\Theta) = \gamma \sum_{c \in \mathcal{Y}} \frac{1}{L_c} \sum_{i=1}^{L} \delta_{y^i=c} \ \log p(\mathbf{x}^i, y^i|\Theta) + (1-\gamma)\frac{1}{U} \sum_{j=L+1}^{L+U} \sum_{c \in \mathcal{Y}} p(y=c|\mathbf{x}^j, \Lambda) \ \log p(\mathbf{x}^j, c|\Theta) \quad (17)$$

where $\delta_{y=c}$ is the Dirac measure equal to 1 if $y = c$ and else 0. We call this variant the *AllClasses* variant since every unlabeled sample contributes to reestimation of the HMMs of all classes.

A smoother version consists to retrain the HMM of a class with its corresponding labeled samples and with unlabeled samples that would be affected to this class by the HCRF system. In this case the objective for learning the whole HMM system writes:

$$\mathcal{L}(\Theta) = \gamma \sum_{c \in \mathcal{Y}} \frac{1}{L_c} \sum_{i=1}^{L} \delta_{y^i=c} \ \log p(\mathbf{x}^i, y^i|\Theta) + (1-\gamma) \sum_{c \in \mathcal{Y}} \frac{1}{\sum_{j=L+1}^{L+U} \delta_{c=\hat{y}^j}} \sum_{j=L+1}^{L+U} \delta_{c=\hat{y}^j} p(c|\mathbf{x}^j, \Theta) \log p(\mathbf{x}^j, c|\Theta) \quad (18)$$

where $\hat{y}^j$ is defined as in Equation 15. We call this variant the *MaxProb* variant since a unlabeled sample will be used to reestimate model of the most likely class only.

Alternatively, instead of weighting the contribution of unlabeled samples by their posterior probability we may simply add samples to the training data set of the HMM of a class according to the HCRF decision. In this case

the objective for learning the whole HMM system writes:

$$\mathcal{L}(\Theta) = \gamma \sum_{c \in \mathcal{Y}} \frac{1}{L_c} \sum_{i=1}^{L} \delta_{y^i=c} \ \log p(\mathbf{x}^i, y^i|\Theta) + (1-\gamma) \sum_{c \in \mathcal{Y}} \frac{1}{\sum_{j=L+1}^{L+U} \delta_{c=\hat{y}^j}} \sum_{j=L+1}^{L+U} \delta_{c=\hat{y}^j} \log p(\mathbf{x}^j, c|\Theta) \tag{19}$$

We call this strategy the *WeightOne* variant.

Finally, in a variant of the *WeightOne* case, one may keep only very likely samples whose conditional probability given by the discriminative model is over a threshold τ (e.g. close to one). In such a case, we are close to a co-training like strategy where a limited number of training samples, labeled by the discriminative model, would be added to the training set of the generative one :

$$\mathcal{L}(\Theta) = \gamma \sum_{c \in \mathcal{Y}} \frac{1}{L_c} \sum_{i=1}^{L} \delta_{y^i=c} \ \log p(\mathbf{x}^i, y^i|\Theta) + (1-\gamma) \sum_{c \in \mathcal{Y}} \frac{1}{\sum_{j=L+1}^{L+U} \delta_{c=\hat{y}^j} \rho_\tau(\hat{y}^j, \mathbf{x}^j)} \sum_{j=L+1}^{L+U} \delta_{c=\hat{y}^j} \rho_\tau(\hat{y}^j, \mathbf{x}^j) \log p(\mathbf{x}^j, c|\Theta) \tag{20}$$

where $\rho_\tau(y, \mathbf{x})$ is equal to 1 if $p(y|\mathbf{x}) \geqslant \tau$ otherwise it is 0. We call this strategy *SelectProb* since it relies on the selection of the likeliest samples.

4.4. Discussion

Our method borrow ideas from co-training and from the work from [40], [48]. Indeed the way the learned HCRF influences the HMM learning is close to the general co-training idea where one model labels samples that

are added to the training set of another classifier. And co-training has been proved efficient in many situations [17] [35] [45]. Moreover, one may note that if the number U of unlabeled data predicted by the discriminative model is much larger than the number of labeled data L, then labeled data may become negligible so that the training set of the generative model follows the distribution given by the discriminative model. Thus, the generative model will tend to be as close as possible to the discriminative one, which we may expect is better than the generative model at previous iteration.

Indeed, the second idea of the algorithm is to learn the discriminative model in a purely supervised way on the labeled training dataset by starting from the HMM solution and regularizing around this initial solution. We can then expect that the HCRF solution will be a local optimum of the conditional likelihood criterion which is close to the HMM system. Being not far from the HMM solution one can expect the HCRF solution to indirectly take into account the unlabeled data. And if the regularization is strong enough one can expect that the HCRF solution will be better than the initial solution, i.e. the HMM system, with respect to the conditional likelihood criterion. Then there are some reason to think that the HCRF solution will be better than the HMM solution.

5. Experiments on artificial data sets

In this section, we illustrate on synthetic data performances of our Iterative Hybrid Algorithm (IHA) compared to benchmark methods : the Entropy Minimization [15] (EM) and the Hybrid Model [40], [48] (HM).

Data are chosen to be as simple as possible in order to enable visual

investigation. We focus on a binary classification where data are generated by two Gaussian distributions (one for each class) with two dimensional feature vectors. The class-conditional densities $p(x|y)$ have the same variance on the y-axis, but are horizontally elongated. Figure 2 illustrates an example of the data distribution.

We investigate the abilities of approaches to learn one isotropic Gaussian distribution per class. This model doesn't capture the horizontal elongation of the true class distributions, so this forms a model mis-specification. Parameters of the model are the means and variances of Gaussian distributions. To simplify the approach, we assume uniform prior probability for each class.

The training data set consists of 200 instances per class, where only few of them are labeled and the testing data set consists of 200 instances per class. Each experiment is run with different random initialization. The parameters of the model are initialized by setting the means of the isotropic Gaussians to the mean of the labeled instances, and setting the variances to one.

We run 150 experiments with 2, 4, and 6 labeled points, 50 runs for each, where in each run we test different values of the hyper-parameters of each method corresponding to the degree of importance of the unlabeled data (e.g. γ in IHA).

First, we investigate results of the Iterative Hybrid Algorithm. Figure 3 shows the performances achieved by the generative and discriminative models (we use the *MaxOne* variant of our approach) as a function of iteration number in three particular runs, for different values of γ, where models are trained on four labeled points during 50 iterations. They show typical behaviors of the method. Although supervised performances are often improved by

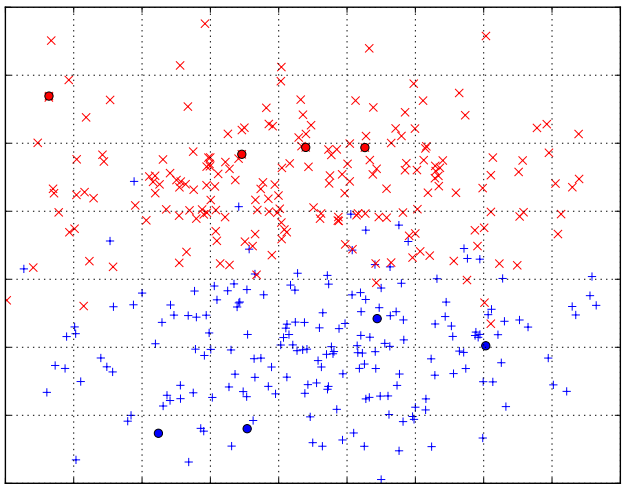


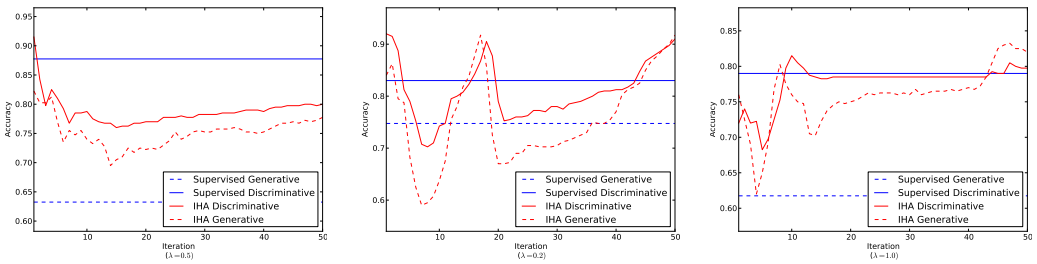
Figure 2: Example of the data distribution with four labeled points per class. The crosses correspond to unlabeled points and the circles correspond to the labeled points.

our iterative framework, performances might be unstable and not always improved from one iteration to another one. For instance, in figure *a* the method starts with a satisfying initial performance, but the performance drops with the number of iterations. In figure *b* the behavior is more chaotic and the accuracy decreases and increases twice, but the final performance achieved is better than the pure discriminative model. Less chaotic behavior is shown in figure *c*, where there are several small fluctuations of the performance, again resulting in better final performance than the pure discriminative model.

Then, we compare the performances of IHA, Hybrid Model and Entropy Minimization in supervised cases (setting hyperparameter values to particular values) and semi-supervised cases (looking for the best value of the hyperparameter). Figure 4 illustrates average performances where for each

method we provide on the left the performance of the supervised case and on the right the performance of the semi-supervised framework reached with the best hyperparameter value. In the Iterative Hybrid Algorithm, $\lambda = 0$ corresponds to a discriminative model trained only on labeled data, but regularized with a generative model trained on both labeled and unlabeled data (see section 2.2). In the Entropy Minimization and the Hybrid Model $\lambda = 0$ and $\alpha = 1$, respectively, correspond to pure discriminative model trained only on labeled data.

In all cases the usage of unlabeled data improves the performance. When there are only two labeled points the best performance is achieved by the Hybrid Model. On four and six labeled points, both the Iterative Hybrid Algorithm and the Hybrid model achieve high performance where the Hybrid Model slightly dominates on four labeled points and the Iterative Hybrid Algorithm dominates when six points are labeled. As it can be seen in the



(a) (b) (c)

Figure 3: Performances of 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.

figure, the difference in the performance for $\lambda = 0$ and the best performance is not very large. This is due to the fact that although the discriminative model does not directly use the unlabeled data, its parameters are regularized with a generative model trained on both labeled and unlabeled data.

In table 1 we take a closer look at the best performance achieved by the methods in each run. We show 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. As in the previous figure, we may notice that on two labeled points the Hybrid Model is dominant. On four labeled points, however, the Iterative Hybrid Algorithm and the Hybrid Model achieve similar performance and

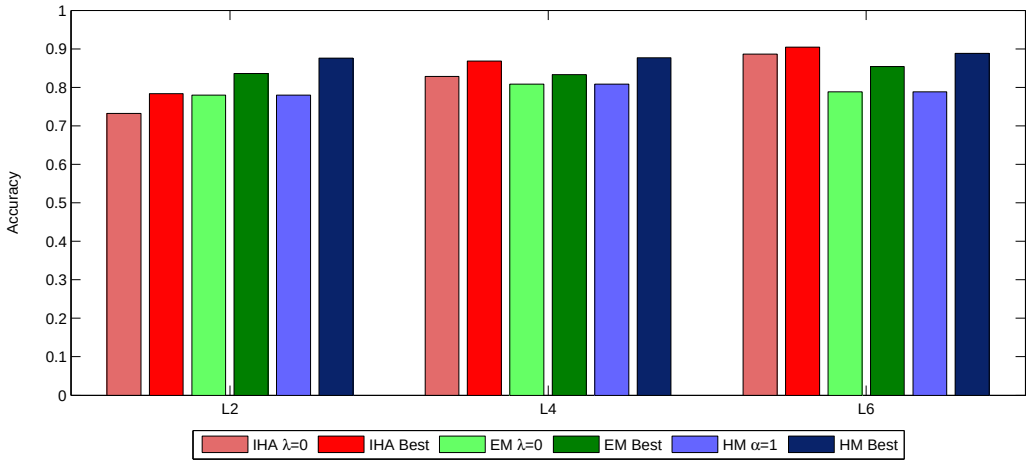


Figure 4: Average performances of the Iterative Hybrid Algorithm (in red), the Entropy Minimization method (in green) and the Hybrid Model (in blue) for supervised cases ($\lambda = 0$ or $\alpha = 1$ cases) and for semi-supervised cases (Best bar). Results are reported for 2 (left), 4 and 6 (right) labeled samples.

L2

	IHA	HM	EM
IHA	0%	16%	36%
HM	84%	0%	68%
EM	64%	24%	0%

L4

	IHA	HM	EM
IHA	0%	40%	72%
HM	52%	0%	82%
EM	28%	14%	0%

L6

	IHA	HM	EM
IHA	0%	72%	96%
HM	22%	0%	72%
EM	2%	16%	0%

Table 1: Percentage of runs in which the method on the left performs better then the method on the top, for 2, 4, and 6 labeled points.

share almost the same percentage of cases in which one outperforms the other. On the other hand, on six labeled points there is clear dominance of the Iterative Hybrid Algorithm, performing better than the Hybrid Model in 72% of the cases and better than the Entropy Minimization in 96% of the cases. In all cases the Entropy Minimization is outperformed by one of the other two methods.

6. Experiments on real datasets

6.1. Datasets and settings

Datasets. We experimented our SSL approach on financial time series and on handwriting data. We present these datasets now.

The financial time series dataset is composed of chart patterns which are particular shape of stock exchange series of interest for financial operators (see Figure 5). We used two databases 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* (and the reverse

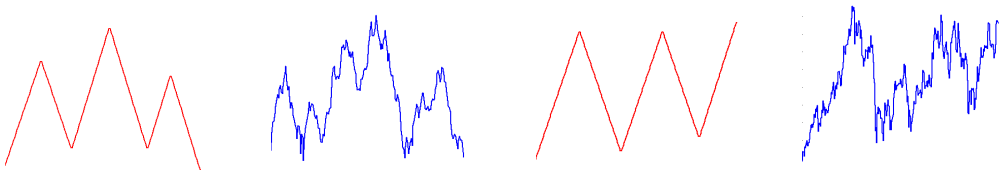


Figure 5: From left to right: ideal shape of a Head and Shoulder pattern (HS), observed HS, ideal shape of an Ascending Triangle pattern (AT), observed AT.

pattern), *Ascending Triangle* and *Descending Triangle*. Datasets are divided into 3 parts : a training dataset with 70 samples per class, a validation dataset and a test dataset with 20 samples per class each.

The handwriting database is a subset of the benchmark IAM database [49] which is made up of images of handwritten letters. The IAM database contains images of English handwriting words which are transformed into series of feature vectors by using a sliding window moving from the left to the right of the image and using preprocessing as in [49] to compute a feature vector from a position of the window. We used two versions of the dataset. A small dataset includes 23 classes and is divided into a training set with 200 samples per class, a validation set and a test set with 50 samples per class each. A bigger dataset 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 validation set and in the test set.

Benchmark methods. We compared our approach with HMMs and HCRFs models trained in supervised setting and with HMMs and HCRFs trained in semi-supervised settings with state of the art approaches. We extend to

HCRFs two main semi-supervised approaches for CRFs which are the semi-supervised method based on entropy minimization [24], and the weighted method proposed in [25]. In the experiments, every HCRF training is performed through two steps: initialization by a HMM system and optimization using stochastic gradient descent on a regularized loss. Also we compared our approach to the general co-training algorithm where the two systems (HMMs and HCRFs) are trained on the same view.

In the following, we will name *HCRF init HMM* a HCRF system initialized by a HMM system, and respectively *weighted HCRF* a semi-supervised HCRF system based on the weighted approach in [25] and *entropy HCRF* a semi-supervised HCRF system based on the approach in [24].

Experimental settings. In all experiments on chart pattern classification the HMM and the HCRF models of every class are left-right models which have either 4 or 6 states depending on the shape of the figures (see Figure 5, e.g. the model of *Head and Shoulders* has six states since its composed of six segments). On IAM dataset all HMM and HCRF models are left-right models with 8 states.

One considered 50 samples of chart pattern training datasets as unlabeled data and for Small IAM and Big IAM datasets one uses 150 and 500 training samples as unlabeled data.

In all experiments, HMMs have a single Gaussian distribution with full covariance matrix as emission probability density.

For strengthening our results, we performed cross validation where folds are build on the training sets. In the following sections we provide first

preliminary results gained with 4 folds cross validation while in the final results (section 6.2.2), we performed 20 folds cross validation on IAM datasets and 60 folds cross validation 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. Thus, in experiments below, the training of each model is performed through a fixed number of iterations, either 4 or 30, this is specified in the text.

6.2. Results

6.2.1. Preliminary results

Iterative framework strategies. We compare first the performances of the variants of our approach as detailed in section 4.3. Table 2 resumes the results.

		Supervised		AllClasses		MaxProb		WeightOne		SelectProb	
database	iterations	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 2: Performances on the test set of supervised HMM and supervised HCRF models compared to different variants of our approach.

The results are reported for 5 labeled data per class whatever the dataset, CP4, CP8 and Small IAM. Supervised models are trained either 4 or 30 iterations while every iteration of our iterative algorithm, models are retrained

for 4 iterations. This table 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 this results confirm comments such as [19], [20] which conclude that this method could sometimes decrease supervised performances. One note also that in most cases running 30 training iterations degrades performances : the labeled dataset size is too small so that models overfit. At the end, one sees that the *MaxProb* strategy significantly outperform the purely supervised training, for both HMM and HCRFs, and seems the best method among all variants, when using only 4 training iterations. We will focus on this variant only for the next experiments.

Labeled datasets sizes. Table 3 reports the performances of supervised HMM and HCRF models compared to our iterative proposal for different number of labeled data per class. We used from 1 to 10 labeled samples per class and 50 or 150 unlabeled samples per class according to the database. One can see that whatever the dataset, CP8 and Small IAM, HCRFs outperform corresponding HMMs and that our approach systematically and significantly outperform the supervised setting.

Evolution of the performances during training. It is interesting to look at the evolution of the performance as a function of the iteration number in our iterative algorithm. Figure 6 plots the performances of HCRF systems and of HMM systems as a function of the iteration number for two iterative

	CP8				Small IAM			
	Supervised		IHA		Supervised		IHA	
labeled data	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%

Table 3: Performances on test set of supervised training for HMMs and HCRFs, compared to semi-supervised training with our approach (*MaxProb* variant) as a function of the number of labeled samples per class, while the number of unlabeled samples remain fixed to 50 samples per class for the CP8 database and 150 samples per class for the Small IAM database.

algorithms, the co-training algorithm and our IHA approach. Note that here one iteration stands for a retraining of both the HMM system and the HCRF system. In our approach this corresponds to a retraining of the HMM systems based on HCRFs classification of unlabeled data and a retraining of the HCRF from the HMM solution. We used here 10 labeled samples and still 50 or 150 unlabeled samples per class for CP8 and Small IAM databases.

We plot as a reference the performance of supervised learning, both for HMMs and for HCRFs. One sees that both iterative algorithms allow improving over supervised training with our proposal being slightly more efficient than standard co-training. The performance of both HMMs and HCRFs increase almost monotonously until it converges. Note also that our approach may reach its best results after few iterations (Small IAM dataset) or may require more iteration (CP8 dataset) to converge to an accurate solution, it

depends on the datasets.

Influence of unlabeled data. At last we investigated the influence of the number of unlabeled data on the performance of our approach. Figure 7 shows the evolution of the accuracy of few systems as the number of unlabeled samples increases while the number of labeled samples remains fixed (5 samples per class). We compare systems learned in a purely supervised settings, semi-supervised systems, i.e. 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 2.2), and systems learned using our iterative approach (note that the semi-supervised systems correspond to the systems gained at the end of the first iteration of our hybrid algorithm). We used 5 labeled samples per class and from 25 to 500 unlabeled samples per class. The first point to note is that SSL systems (both HMMs and HCRFs) outperform the corresponding supervised systems and that the accuracy increases up to a plateau. The second point is that the iterative algorithm allows learning even more accurate classifiers, both HMMs and HCRFs, and that their performance increases steadily with the number of unlabeled data.

6.2.2. Comparative results with state of the art methods

Finally, we compared more extensively our methods with more state of the art semi-supervised methods on our four datasets (CP4, CP8, small IAM and Big IAM). In these experiments to provide more significant results we report averaged results gained with 20 folds cross validation on the IAM corpus and with 60 folds cross validation on Chart Pattern databases. This allows providing 95% confidence interval. Note that since standard semi-

supervised training of HMM do not always improve over the supervised case, we investigated different variants of the standard algorithm and obtained that semi-supervised HMMs offers best performances when using the *SelectProb* and the *MaxProb* strategies as in equations 20 and 18, we report here best results following datasets that is *MaxProb* strategy for Chart Patterns and *SelectProb* for IAM datasets. Also, following preliminary results, we chose to use the *MaxProb* strategy for our iterative framework. All models are trained on 5 labeled samples per class and we use 50 unlabeled samples per class on the Chart Pattern datasets, 150 and 500 unlabeled samples per class on the Small IAM and Big IAM datasets.

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	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 HCRF entropy	84.0% ±0.6	62.0% ±0.9	37.6% ±0.9	43.2% ±1.0
SSL HCRF weighted	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

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.

This table calls from a few comments. First SSL learning systematically

outperform supervise learning, for both HMM and HCRF systems. Second, entropy SSL entropy and weighted SSL are very close whatever the dataset, and yield small improvement over simple SSL training for HMMs and for HCRFs, if any. Co-training performs sometimes better, and sometimes worse than simple SSL methods, and appears to be less robust and maybe more difficult to tune. Finally, our Iterative Hybrid Algorithm most often outperforms all other methods, both for HMM and for HCRF systems. Note that although improvements are often small, they are bigger on the IAM data set than on the CPx datasets and they are systematic.

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

7. Conclusion

We presented a joint HMM-HCRF framework for semi-supervised learning of graphical models for sequences. Our approach combines on the one hand the initialization scheme of HCRFs system by HMMs for learning a HCRF model using unlabeled data and on the other hand a co-training procedure for improving a HMM system based on a HCRF model. Our experimental results on two 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 applied to HCRF learning and to the well known co-training algorithm.

References

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

[2] B. Juang, S. Katagiri, Discriminative learning for minimum error classification, in: IEEE Trans. Acoustics, Speech, and Signal Processing, Vol.40, No.12, 1992.

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

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

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

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

[7] J. D. Lafferty, A. McCallum, F. C. N. Pereira, 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, 2001, pp. 282–289.

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

[9] S. Reiter, B. Schuller, G. Rigoll, Hidden conditional random fields for meeting segmentation, in: ICME, 2007, pp. 639–642.

[10] T.-M.-T. Do, T. Artières, 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, 2005.

[11] T.-M.-T. Do, T. Artières, 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), 2006.

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

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

[14] Y. Soullard, T. Artieres, 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, 2011, pp. 453–458.

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

[16] J. Wang, X. Shen, W. Pan, On efficient large margin semisupervised learning: Method and theory, Journal of Machine Learning Research 10 (2009) 719–742.

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

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

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

[20] M. Inoue, N. Ueda, Exploitation of unlabeled sequences in hidden markov models, IEEE Trans. On Pattern Analysis and Machine Intelligence 25 (2003) 1570–1581.

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

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

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

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

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

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

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

[28] G. S. N. W. L. E. Baum, T. Petrie, A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains (1970) 164–171, vol 41, No 1.

[29] L. Bottou, On-line learning in neural networks, Cambridge University Press, New York, NY, USA, 1998, Ch. On-line learning and stochastic approximations, pp. 9–42.

[30] D. C. Liu, J. Nocedal, D. C. Liu, J. Nocedal, On the limited memory bfgs method for large scale optimization, Mathematical Programming 45 (1989) 503–528.

[31] F. Sha, Large margin training of acoustic models for speech recognition, Doctoral dissertation (2006).

[32] U. Brefeld, T. Scheffer, Semi-supervised learning for structured output variables, in: ICML, 2006, pp. 145–152.

[33] X. Zhu, Semi-Supervised Learning Literature Survey Contents, Sciences-New York.

[34] S. Baluja, Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data, Advances in Neural Information Processing Systems 11 (1998) 854–860.

[35] W. Wang, Z. hua Zhou, Analyzing co-training style algorithms, in: Proceedings of the 18th European Conference on Machine Learning, 2007.

[36] D. Zhou, J. Huang, B. Scholkopf, Learning from Labeled and Unlabeled Data on a Directed Graph, Proceedings of the 22nd International Conference on Machine Learning (2005) 1041–1048.

[37] A. Blum, S. Chawla, Learning from Labeled and Unlabeled Data using Graph Mincuts, in: Science, Morgan Kaufmann Publishers Inc., 2001, pp. 19–26.

[38] E. Riloff, J. Wiebe, Learning Extraction Patterns for Subjective Expressions, in: E. Riloff, J. Wiebe (Eds.), Proceedings of the Conference on

Empirical Methods in Natural Language Processing EMNLP, Conference on Empirical Methods in Natural Language Processing, 2003, pp. 105–112.

[39] D. Yarowsky, Unsupervised word-sense disambiguation rivalling supervised methods, in: 33rd Annual Meeting of the Association for Computational Linguistics, 1995, pp. 189–196.

[40] J. L. Christopher M. Bishop, Generative or discriminative? getting the best of both worlds (2006) 3–24.

[41] G. Bouchard, 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, 2007, pp. 124–129.

[42] S. Ji, L. T. Watson, L. Carin, Semisupervised learning of hidden markov models via a homotopy method., IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 275–287.

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

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

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

[46] V. Frinken, T. Peter, A. Fischer, H. Bunke, T.-M.-T. Do, T. Artieres, 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, 2009, pp. 189–196.

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

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

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

Appendix A. HCRF initialization from HMMs

We consider here HMMs with one Gaussian distribution with full covariance matrix as emission probability density. We show how a HMM of Q states may be used to initialize a HCRF with the same topology.

We note for any state $i \in [1, Q]$, $\boldsymbol{\mu}^i$ and Σ^i the mean and covariance matrix of the gaussian distribution in state i. Let d the dimension of feature vectors, then $\boldsymbol{\mu}^i$ is a vector of dimension d and Σ^i is a matrix of dimension

$d \times d$. Then, we define the following feature maps and parameter vectors:

$$\boldsymbol{\phi}^{trans}(\mathbf{x}, y, h_t, h_{t-1}) = \left(\delta_{h_t=1 \wedge h_{t-1}=1}, \ldots, \delta_{h_t=Q \wedge h_{t-1}=Q}\right)^T$$
$$\boldsymbol{\lambda}^{trans} = \left(\log a_{1,1}, \ldots, \log a_{Q,Q}\right)^T$$
$$\boldsymbol{\phi}^{loc}(\mathbf{x}, y, h_t) = \left(\boldsymbol{\phi}_1^{loc}(\mathbf{x}, y, h_t), \boldsymbol{\phi}_2^{loc}(\mathbf{x}, y, h_t), \ldots, ..., \boldsymbol{\phi}_Q^{loc}(\mathbf{x}, y, h_t)\right)^T$$
$$\boldsymbol{\lambda}^{loc} = \left(\boldsymbol{\lambda}_1^{loc}, \boldsymbol{\lambda}_2^{loc}, \ldots, ..., \boldsymbol{\lambda}_Q^{loc}\right)^T$$

where $a_{i,j}$ stands for the usual HMM transition probability from state i to j.

Using these definition, we get easily that if $\boldsymbol{\phi}^{loc}$ and λ^{loc} satisfy:

$$\left\langle \boldsymbol{\lambda}_i^{loc}, \boldsymbol{\phi}_i^{loc}(\mathbf{x}, y, s_t) \right\rangle = (\mathbf{x} - \boldsymbol{\mu})^T (\Sigma^i)^{-1} (\mathbf{x} - \boldsymbol{\mu}) - \frac{1}{2} log((2\pi)^d |\Sigma^i|) \qquad \text{(A.1)}$$

then:

$$e^{\boldsymbol{\lambda}_i^{loc} . \boldsymbol{\phi}_i^{loc}(\mathbf{x}, y, h_t)} = \frac{1}{\sqrt{(2\pi)^d |\Sigma^i|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T (\Sigma^i)^{-1} (\mathbf{x}-\boldsymbol{\mu})} \qquad \text{(A.2)}$$

It is easy to check that the definitions below allow satisfying the equality above:

$$\boldsymbol{\phi}_i^{loc}(\mathbf{x}, y, h_t) = \left(1, (x_u)_u, (x_u x_v)_{u,v}\right)^T \times \delta_{h_t=i} \qquad \forall i \in [1, Q]$$

$$\boldsymbol{\lambda}_i^{loc} = \begin{pmatrix} -\frac{1}{2}\left[\log((2\pi)^d |\Sigma^i|) + (\boldsymbol{\mu}^i)^T (\Sigma^i)^{-1} \boldsymbol{\mu}^i\right] \\ \left((\Sigma^i)^{-1} \boldsymbol{\mu}^i\right) \\ \left(-\frac{1}{2} (\Sigma^i)^{-1}_{u,v}\right)_{u,v} \end{pmatrix} \forall i \in [1, Q]$$

where we use the notation:

$$(x_u)_u = \mathbf{x} = (x_1, ..., x_d)^T$$
$$(x_u x_v)_{u,v} = \mathbf{x} \otimes \mathbf{x} = \left(x_1^2, x_1 x_2, ..., x_1 x_d, x_2 x_1, ..., x_d x_1, ..., x_d^2\right)^T$$

In previous derivation $A_{u,v}$ stands for the element line u and column v of the matrix A. Also, $(A_{u,v})_{u,v} = \left(A_{1,1}, A_{1,2}, \ldots, A_{2,1}, A_{2,2}, \ldots, A_{3,1}, \ldots\right)^T$ is a vector of the elements of A unfolded in column first order.

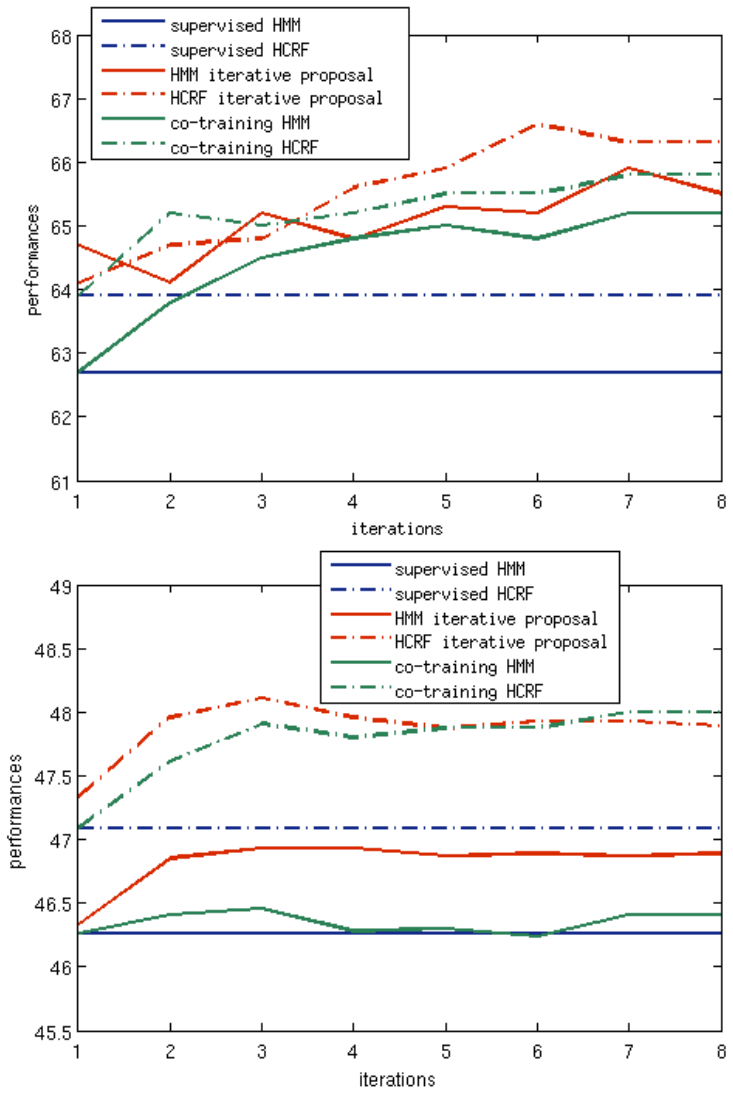


Figure 6: Performance on the test set of HMMs and HCRFs models following iteration number in our iterative hybrid algorithm for CP8 (top) and small IAM (bottom) datasets.

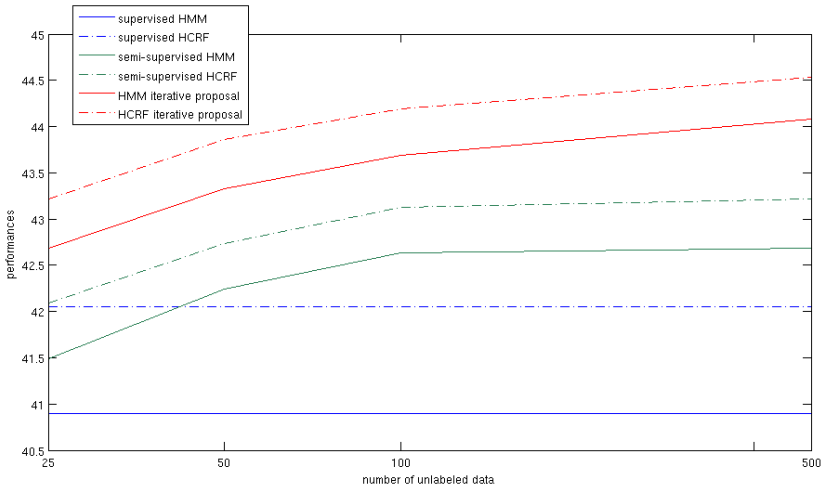


Figure 7: Comparison of the accuracy of HMM and HCRF systems trained in a supervised and in a semi-supervised setting with HMM and HCRF systems learned with our iterative approach on the Big IAM dataset. Performance is plotted as a function of the number of unlabeled samples used.