VARIATIONAL SEQUENCE LABELING

R. Flamary, S. Canu, A. Rakotomamonjy

LITIS EA 4108, Université de Rouen
76800 Saint Etienne du Rouvray France

J.L. Rose

CREATIS-LRMN, Université de Lyon
69621 Villeurbanne France

ABSTRACT

Sequence labeling is concerned with processing an input data sequence and producing an output sequence of discrete labels which characterize it. Common applications includes speech recognition, language processing (tagging, chunking) and bioinformatics. Many solutions have been proposed to partially cope with this problem. These include probabilistic models (HMMs, CRFs) and machine learning algorithm (SVM, Neural nets). In practice, the best results have been obtained by combining several of these methods. However, fusing different signal segmentation methods is not straightforward, particularly when integrating prior information. In this paper the sequence labeling problem is viewed as a multi objective optimization task. Each objective targets a different aspect of sequence labelling such as good classification, temporal stability and change detection. The resulting optimization problem turns out to be non convex and plagued with numerous local minima. A region growing algorithm is proposed as a method for finding a solution to this multi functional optimization task. The proposed algorithm is evaluated on both synthetic and real data (BCI dataset). Results are encouraging and better than those previously reported on these datasets.

1. INTRODUCTION

The problem we propose to deal with is the one of sequence labeling. In a signal classification context, the aim of sequence labeling would consist in obtaining a label for each sample of the signal while taking into account the sequentiality of the samples by imposing some temporal constraints on the labeling. For instance, such constraints can be posed on the length of segments of similar label or on the time neighborhood. Problems of sequence labeling typically arise in speech signal segmentation or in Brain Computer Interfaces (BCI). Indeed, for BCI applications, classifying mental tasks using EEG continuous signals are often needed. This consists in assigning a label to each sample (or set of samples) of the EEG signal and afterwards each label can be interpreted as a specific command for the BCI application.

Many methods and algorithms have already been proposed for sequence labeling. For instance, Hidden Markov Models [1] is a statistical method that is able to learn a joint probability distribution of the samples of a sequence and their labels. In a sequence labeling problem, Conditional Random Fields (CRF) [2] learn instead of a generative model, conditional probability of the labels given the samples. In some cases, CRF have been shown to improve the HMM approach as they do not need the observations to be independent. Structural SVMs that learn a mapping from structured input to structured output have also been considered for sequence labeling [3]. In the same flavor, Maximum Margin Markov Networks [4] learn a probabilistic graphical model that, after training, is able to infer the labels associated to a sequence samples.

All these methods have been compared by Nguyen and Guo [5] who proposed a method for combining the labeled sequence returned by the above-described approaches. Their combination method consists in obtaining an optimal sequence $\mathbf{y}^*$ given the results of several sequence labeling methods $(\mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^M)$:

$$\mathbf{y}^* = \arg\min_{\mathbf{y}} \mathcal{L}(\mathbf{y}, \mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^M) \tag{1}$$

with $\mathcal{L}$ a loss function that takes into account the label provided by each method and all label transitions. They have shown that such an optimally fused combination of sequence labels is consistently better than a sequence label obtained by a single method. This novel method makes possible the combination of several sequence labeling obtained from different methods. However, most sequence labeling methods provide a score of each class for a given sample and the label is then obtained by :

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} f(X, \mathbf{y}) \tag{2}$$

with f a score function depending on the current sequence $\mathbf{y}$ and the observation X. For instance, structural SVM produces a score of the form $f(X, \mathbf{y}) = w^t \Phi(X, \mathbf{y})$. For combining the outputs of several sequence labeling methods, similarly to methods in soft-decision decoding ([6] Chapter 7), we propose to directly use the scores instead of the output labels. As detailed in the sequel, this would lead us to minimize a loss function which depends on the scores returned by different labeling methods. In our case, we minimize a sum of functionals each one corresponding to a specific method. Furthermore, through this formulation, we can integrate prior knowledge by adding a functional related to such a knowledge to the sum, leading to a more difficult problem. However it can be solved using a variational approach. This means that instead of considering functional values, we consider their variation with respect to a change of label for a given sequence. Recently, Rose et al.[7] has cast the problem of labeled image segmentation as the minimization of a sum of functionals. The minimization problem was solved through a Region Growing algorithm.

Thus, our main contribution in this paper is to propose a variational framework for combining different sequence labeling methods and prior knowledge. We propose to introduce a region growing like algorithm denoted as variational Sequence Labeling Algorithm for solving the resulting optimization problem.

The paper is organized as follows. First, we will express our problem as a sum of functionals, each of them bringing information about the problem and accuracy to the final solution. We propose several functionals of which some corresponds to existing methods cited above. Then, we will present our algorithm to solve our minimization problem. Finally, we will test our method on a toy dataset and on a BCI Mental Task segmentation.

2. VARIATIONAL SIGNAL SEGMENTATION

For learning, we assume to have a sequence X^{tr} of length T^{tr} where each sample belongs to $\mathbb{R}^D$ and a sequence of labels $\mathbf{y}^{tr} \in \{1,..,N\}^{T^{tr}}$. We suppose that X^{tr} is gathered as matrix of dimension $T^{tr} \times D$. For testing, we have a sequence X of length T. We denoted as $X_i \in \mathbb{R}^D$ the ith sample of sequence X.

2.1. Variational Approach

We propose to cast our problem in the context of variational framework:

$$\min_{\mathbf{y}} \sum_{i=1}^{N_f} \lambda_i J_i(\mathbf{y}, X, \mathbf{y}^{tr}, X^{tr}) \tag{3}$$

with each functional J_i is balanced by $\lambda_i \in \mathbb{R}^+$.

Typically in clustering, a criterion that maximizes the similarity of the samples in each class is generally used. A valid functional, in this case, is the sum for every sample of the distance intra-classes. But if we want to integrate a prior knowledge, a second functional corresponding to this information should be added. The arguments given to a functional depends on the criterion. For example, a supervised learning would imply a functional using $(X, X^{tr}, \mathbf{y}^{tr})$, the training and test set. Whereas a functional for a prior information like the number of regions would use only $\mathbf{y}$.

Variational methods have been used in the context of supervised learning [8] and in image segmentation [7]. For instance in the latter work, Rose et al. have integrated a shape prior information with a region-based criterion. In this work, we propose to adapt this variational approach to sequence labeling. In the next section, we define functionals derived from existing methods.

2.2. Labeling functional

This functional is the one corresponding to a supervised learning criterion. For that, we need to obtain N decision functions returning a score for a sample to be a member of class n.

$$f_n = \arg\min_f \; \mathcal{L}_n(\mathbf{y}^{tr}, f(X^{tr})) + \lambda\Omega(f) \tag{4}$$

with $\mathcal{L}_n$ a loss function for class n and Ω a regularization term. In this work, we propose to use the following general form for the labeling functional:

$$J_{class}(\mathbf{y}, X) = -\sum_{i=1}^{T} f_{\mathbf{y}_i}(X_i) \tag{5}$$

By minimizing this functional, we choose for each sample the class with the maximum score:

$$\begin{aligned} \min_{\mathbf{y}} J_{class}(\mathbf{y}, X) &\equiv \max_{\mathbf{y}} \sum_{i=1}^{T} f_{\mathbf{y}_i}(X_i) \\ &\equiv \sum_{i=1}^{T} \max_{\mathbf{y}_i} f_{\mathbf{y}_i}(X_i) \end{aligned} \tag{6}$$

That corresponds to a Winner-Takes-All strategy: we would choose for $\mathbf{y}$ classes having the maximum score for each sample (Fig. 1). We can see that any machine learning algorithm returning a score may be used. But it seems sensible to adapt this functional to specific cases, for instance when the value returned by f_n is a membership probability to class n ($f_n(X_i) = P(\mathbf{y}_i = n|X_i)$). If we suppose that the samples are independent, then :

$$\begin{aligned} \max_{\mathbf{y}} P(\mathbf{y}|X) &= \max_{\mathbf{y}} \prod_{i=1}^{T} P(\mathbf{y}_i|X_i) \\ &\equiv \max_{\mathbf{y}} \sum_{i=1}^{T} \log(P(\mathbf{y}_i|X_i)) \end{aligned} \tag{7}$$

So the functional corresponding to the maximization of likelihood can be formulated as:

$$J_{classp}(\mathbf{y}, X) = -\sum_{i=1}^{T} \log P(\mathbf{y}_i|(X_i)) \tag{8}$$

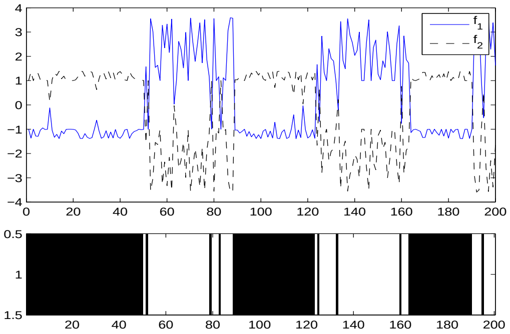

Fig. 1. Score returned by f_n functions over time. The bottom plot shows the estimated class obtained by WTA strategy (black is label 2 and white is label 1)

We defined two functionals corresponding to a supervised classification criterion. If used alone, these functionals boils down to usual supervised machine learning approach. But as we can see in Figure 1, if the score is noisy over time there are misclassified samples. Next, we propose new functionals that can be seen as regularizing terms that may smooth the sequence.

2.3. Change detection functional

In machine learning community, several methods to detect class changes along time have been proposed. For instance Desobry et al [9] introduced the kernel change detection (KCD) method. If we have information concerning label changes along time, we should use it in the sequence labeling process. So we propose a functional that takes into account this kind of information.

First we define f_c a function returning a change detection score over time. An example of a returned change score over time is showed in Figure 2. We consider only the class changes in the segmentation. So we define a function, *edge* that returns a list of the indexes of the changes for a given **y**:

$$L_c = edge(\mathbf{y}) = \{i : |\mathbf{y}_{i+1} - \mathbf{y}_i| \neq 0\} \in \mathbb{N}^{N_c} \qquad (9)$$

where L_c is the list of changes and N_c the number of changes. We propose a functional that maximizes the changes position precision:

$$J_{edge}(\mathbf{y}, X) = -\frac{1}{N_c} \sum_{i \in L_c} f_c(X_i), \qquad (10)$$
$$\text{with } L_c = edge(\mathbf{y}) \in \mathbb{N}^{N_c}.$$

Minimizing this functional move the borders of the regions to positions with the highest change score. Note that the functional is divided by the number of changes because otherwise a trivial solution would be to have a change for each samples if $f_c > 0$.

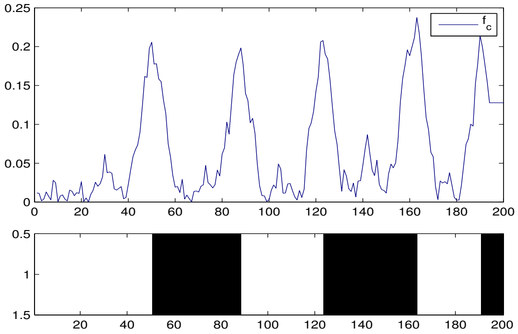

Fig. 2. Example of score returned by a change detection algorithm over the time. The bottom plot show how the label of the signal samples are structured. White and black regions respectively correspond to samples of class 1 and 2. The upper plot depicts the evolution of the score. We can note that the score becomes larger around regions where sample labels change.

2.4. Prior information functionals

Now, we propose to define new functionals corresponding to a prior information criterion. These functionals are important as they force the solution to have some properties, for instance in sequence labeling, we may suppose that signal segments of same label have a minimal length.

Total variation functional

When we know that the solution has large regions, it may be useful to limit the number of change by minimizing this functional:

$$J_{TV}(\mathbf{y}) = \sum_{i=1}^{T-1} \|\mathbf{y}_{i+1} - \mathbf{y}_i\|_0 \qquad (11)$$

where $||.||_0$ is the ℓ_0 norm.This functional has been widely used in signal processing and image processing community and is called Total Variation (TV).

Markov Model functional

The second prior information that we propose is one based on transition probabilities between classes. If we use a Markov Model to describe classes changes, the model is defined by a transition matrix M:

$$M(c1, c2) = P(\mathbf{y}_i = c2 | \mathbf{y}_{i-1} = c1) \qquad (12)$$

This M matrix can be either estimated using the training set or defined by a user having an prior knowledge about the segmented process.

The functional corresponding to a Markov Model with a transition matrix M is:

$$\begin{aligned} J_{MM}(\mathbf{y}, M) &= -\sum_{k=1}^{T-1} \log P(\mathbf{y}_{k+1}|\mathbf{y}_k) \\ &= -\sum_{k=1}^{K-1} \log M(\mathbf{y}_k, \mathbf{y}_{k+1}) \end{aligned} \tag{13}$$

This functional enables to have an *a priori* concerning the proportion of each class in the final segmentation and in the sequence of classes.

2.5. Hidden Markov Models

In this section, we discuss the link between our framework and the existing signal segmentation method known as Normal HMM. HMM are based on the maximization of the probability of having a given sequence knowing an observation. So, if we want to put the HMM framework in our variational one, we have to minimize the functional:

$$J_{HMM}(\mathbf{y}, X) = -P(\mathbf{y}|X) \tag{14}$$

which is equivalent to:

$$\begin{aligned} J'_{HMM} &= -\log P(\mathbf{y}|X) \\ &= -\log\left(\prod_{i=1}^{T} P(\mathbf{y}_i|X_i) \prod_{i=2}^{T} P(\mathbf{y}_i|\mathbf{y}_{i-1})\right) \\ &= -\sum_{i=1}^{T} \log P(\mathbf{y}_i|X_i) - \sum_{i=1}^{T-1} \log M(\mathbf{y}_i, \mathbf{y}_{i+1}) \\ &= J_{classp}(\mathbf{y}, X) + J_{MM}(\mathbf{y}, M) \end{aligned} \tag{15}$$

where M is the transition matrix of the HMM.

The HMM is a special case of our framework as it corresponds to the minimization of the sum of two functionals defined section 2. This method known for its efficiency in sequence labeling uses two functionals: the first one is the data term and the second one is the regularization term. But using HMM implies λs coefficients to be equal and do not permit the user to balance between data precision and model precision.

For the HMM continuous case (Normal HMM[1]), the observations X have been generated by a known mixture of Gaussian depending on the hidden state. The algorithm used to obtain the HMM model is the EM algorithm and the Viterbi algorithm is used to obtain the most probable sequence.

HMM does not provide easy integration of prior information. In our variational framework, one can easily add prior information to HMM functionals to improve the result. For instance J_{edge} may improve the result as it force the transitions to be on the detected label change.

3. ALGORITHM

Current existing methods for sequence labeling are provided with efficient ad hoc algorithms. But these algorithms can not be used for any sum of functionals. In this context, we propose to use an algorithm based on a commonly used method in image processing: Region Growing. This algorithm is iterative, first an initial sequence is set, then the borders of the regions will be moved depending on a criterion.

Algorithm 1 Variational Sequence Labeling Algorithm (VSLA)

```
Initialization of $\mathbf{y}^0$
for $i = 1, 2, \cdots$ do
  $L_c = edge(\mathbf{y}^{i-1})$ {Edge moving loop}
  for $j \in L_c$ do
    $\mathbf{y}^+ = \mathbf{y}^{i-1}; y_j^+ = \mathbf{y}_{j-1}^{i-1}$
    $\mathbf{y}^- = \mathbf{y}^{i-1}; y_j^- = \mathbf{y}_{j+1}^{i-1}$
    $\Delta J^+ = J(\mathbf{y}^+) - J(\mathbf{y}^{i-1})$
    $\Delta J^- = J(\mathbf{y}^-) - J(\mathbf{y}^{i-1})$
    $c = \arg\min[\Delta J^- \ 0 \ \Delta J^+]$
    $V = [\mathbf{y}^- \ 0 \ \mathbf{y}^+]$
    $\mathbf{y}^i = V(:, c)$
  end for
  $L_r = region(\mathbf{y}^i)$ {Region changing loop}
  for $j = 1 \ldots |L_r|$ do
    I= indexes or region $L_r(k)$
    for $k = 1 \ldots N$ do
      $\mathbf{y}^* = \mathbf{y}^i; \mathbf{y}_I^* = k$
      $\Delta J_k = J(\mathbf{y}^*) - J(\mathbf{y}^i)$
    end for
    $c = \arg\min \mathbf{\Delta J}$
    $\mathbf{y}_I^i = c$
  end for
  if stopping criterion then
    break
  end if
end for
with $region$ a function returning the list of all regions for
a given segmentation.
```

Our algorithm (Algo. 1), is iterative and two steps will be performed at each iteration. First, every edge in the current $\mathbf{y}$ will be tested and moved depending on the variation of $J(\mathbf{y})$. This step corresponds to the classical Region Growing Algorithm. Then every region of the current $\mathbf{y}$ will be called into question and their class will be changed depending on the variation of $J(\mathbf{y})$. This step was added in order to speed up the sequence labeling process. Note that the calculation of $J(\mathbf{y})$ is in fact never done. For each border movement only the variation ΔJ is processed. Simple algebras show that ΔJ may be found in a computationally efficient way. For instance the variation of J_{class} for changing the class of the ith sample from c_1 to c_2 is:

$$\Delta J_{class}(X, i, c_1, c_2) = f_{c_1}(X_i) - f_{c_2}(X_i) \qquad (16)$$

In the same way, an efficient variation for a complete region changing may be found.

The number of iteration before convergence is strongly dependent on the initialization. Then, we have to initialize our sequence at a value near from the optimum. In order to obtain an initialization $\mathbf{y}$ near the optimum, we solve a simpler version of our sum of functionals. For instance the initialization is the optimum $\mathbf{y}$ so that J_{class} is minimized, which is just an element-wise minimization.

4. RESULTS

4.1. Toy dataset

We built a toy problem based on a 1D nonlinear multi-class switching mean. For each class two values are equally probable possible and the classes are changing over time. A Gaussian noise is then added to this generated signal. Figure 3 shows an example of a 3 classes toy dataset and his estimated normal densities ($p(x|l)$).

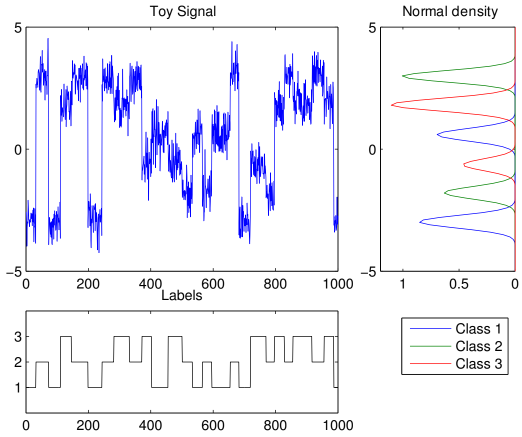

Fig. 3. Toy Dataset Example for 3 classes: each class can be of 2 values and their estimated(EM) class-conditional densities are shown on the right. Below, the class corresponding to the signal can be seen

A training set of 2000 points, a validation set of 2000 points and a test sets of 4000 points were generated. A validation method was used to choose the values of the λ_i coefficients (brute force).

We used several classification methods to obtain scores: SVM multi-class classifier, Mixture of Gaussian Classifier (MG) and Kernel Ridge Regression (KRR). For each method, we considered the J_{class} functional alone and with other functionals. We measured the accuracy of the segmentation on the test set for 10 generated datasets and the averaged results can be seen in Table 1.

Fusing scores obtained from different classification methods (SVM+MG+KRR) does not improve the result in this case. The problem is one-dimensional and simple enough to obtain robust classifiers from many methods. Hence, all classifiers give similar scores and do not provide enough diversity.

	SVM		MG	
	J_{class}	$+J_{edge}$	J_{classp}	$+J_{edge}$
$\varnothing$	0.7111	0.7174	0.7393	0.7400
$+J_{TV}$	0.8677	**0.8741**	**0.9311**	0.9289
$+J_{MM}$	0.8138	0.8104	0.9005	0.9002

	KRR	
	J_{class}	$+J_{edge}$
$\varnothing$	0.7343	0.7480
$+J_{TV}$	0.9155	**0.9189**
$+J_{MM}$	0.8775	0.8844

Table 1. Results for the Toy Dataset

We can see that surprisingly, with our algorithm, the Markov Model functional does not have results as good as the Total Variation one. It comes from the fact that the toy example does not have a strong prior concerning label transition. Moreover J_{edge} functional does not always bring a better accuracy to the global solution.

4.2. BCI dataset

We used the Dataset from BCI Competition III provided by J. del R Millãn[10] to test our method. The problem was to determine the mental state of a subject along time. Three mental state were possible: subject thinking about his left hand, his right hand and his foot.

The features proposed in the competition are the Power Spectral Density of the EEG electrodes. And the best result were obtained by Linear Discrimination. To test our approach we used the given features without preprocessing and the classification method was a linear regression with canal selection[11].

We have 3 sessions for the training and 1 session for the test. The third training session was used as validation set to obtain the best values of the λ_i parameters.

Functionals	Subject 1	Subject 2	Subject 3
J_{class}	0.7392	0.6262	0.4931
$\ldots + J_{TV}$	**0.9843**	**0.8531**	**0.5932**
$\ldots + J_{MM}$	0.9783	0.7955	0.4455
BCI III Res.	0.9598	0.7949	0.6743

Table 2. Results for the BCI Dataset

Results obtained can be seen on Table 2. Depending on the subject we obtain better accuracy than the best competition result. In fact, we see clearly in this application that the final result strongly depends on efficiency of the classification. The best example is the subject 3 because its accuracy using only J_{class} is only 49% which is not far from the classification made by chance (33%), we couldn't improve the results on this subject because the original classification was not good enough.

Matlab code corresponding to these results will be provided on our website.

5. CONCLUSION

In this paper, we proposed a novel method for combining several sequence labeling methods which allows us to integrate to the sequence labeling problem some prior known information. For this purpose, we expressed our problem as a weighted multi-objective problem. Owing to this framework, integrating prior information on the problem can simply be done by adding a functional to the optimization objective function. The resulting minimization problem being difficult, we proposed an algorithm inspired from Region Growing approach used in image segmentation.

We tested our method on several examples. The first one is a toy dataset corresponding to the sequence labeling of a noisy signal. The second one is a BCI mental task segmentation problem proposed in the BCI competition III. Our results show that our method is promising and competitive with respect to the state of the art in particular for the BCI Dataset.

Similarly to what we have shown for the Normal HMM, we believe that many of the currently known sequence labeling methods such as CRF or stuctural SVM can be expressed in a variational framework and thus can be used within the method we proposed. Our future work will address such a point in order to obtain new functionals bringing others prior information or data information.

6. REFERENCES

[1] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*, Springer, 2005.

[2] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th International Conf. on Machine Learning*, 2001, pp. 282–289.

[3] I. Tsochantaridis, J. Thorsten, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," in *Journal Of Machine Learning Research*, Cambridge, MA, USA, 2005, vol. 6, pp. 1453–1484, MIT Press.

[4] B. Taskar, C. Guestrin, and D. Koller, "Max-margin markov networks," in *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004, MIT Press.

[5] N. Nguyen and Y. Guo, "Comparisons of sequence labeling algorithms and extensions," in *Proc. 24th international Conf. on Machine learning*. 2007, pp. 681–688, ACM.

[6] Robert H. Morelos-Zaragoza, *The Art of Error Correcting Coding*, Wiley, 2006.

[7] J-L Rose, C Revol-Muller, C Reichert, and C Odet, "Variational region growing," *VISAPP 2009 Proceedings*, 2009.

[8] K.R. Varshney and A.S. Willsky, "Supervised learning of classifiers via level set segmentation," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2008.

[9] F. Desobry, M. Davy, and C. Doncarli, "An online kernel change detection algorithm," *IEEE Transactions on Signal Processing*, vol. 53, pp. 2961–2974, Aug. 2005.

[10] J. del R Millán, "On the need for on-line learning in brain-computer interfaces," in *Proc. Int. Joint Conf. on Neural Networks*, 2004.

[11] A. Rakotomamonjy, "Algorithms for multiple basis pursuit denoising," in *Workshop on Sparse Approximation*, 2009.