SEQUENTIAL LABELING WITH STRUCTURAL SVM UNDER THE $F_1$ LOSS

Guopeng Zhang and Massimo Piccardi

School of Computing and Communications, University of Technology, Sydney, Australia

ABSTRACT

Sequential labeling addresses the classification of sequential data and is of increasing importance for the classification and segmentation of video data. The model traditionally used for sequential labeling is the hidden Markov model where the sequence of class labels to be predicted is encoded as a Markov chain. In recent years, hidden Markov models and other structural models have benefited from minimum-loss training approaches which in many cases lead to greater classification accuracy. However, the loss functions available for training are restricted to decomposable cases such as the zero-one loss and the Hamming loss. Other useful losses such as the $F_1$ loss, equal error rates and others are not available for sequential labeling. For this reason, in this paper we propose a training algorithm that can cater for the $F_1$ loss and any other loss function based on the contingency table. Experimental results over the challenging TUM Kitchen Dataset depicting human actions in a kitchen scenario show that the proposed training approach leads to significant improvement of different performance metrics such as the classification accuracy (4.3 percentage points) and the $F_1$ measure (8.9 percentage points).

Index Terms— Sequential labeling, hidden Markov model, structural SVM, Hamming loss, $F_1$ loss.

1. INTRODUCTION

Sequential labeling has been drawing increasing attention from the image processing and machine learning communities [1, 2, 3, 4, 5]. Also known as tagging, or decoding, sequential labeling consists of the assignment of a sequence of class labels to a sequence of measurements. This problem is a fundamental problem in many image and video processing applications such as image segmentation, handwriting recognition, activity recognition, pose estimation, and also in fields as diverse as gene finding, protein secondary structure prediction, financial analysis and many others.

A simple, yet effective, model used for sequential labeling is the hidden Markov model (HMM) [6]. An HMM consists of two sets of variables: a sequence of class variables linked in a first-order Markov chain, and a sequence of measurements each depending only on the corresponding class variable. The parameters of this model have traditionally been learned by maximising the likelihood function over a set of supervised examples. However, in recent years other styles of training have become increasingly popular; in particular, the extension of the support vector machine (SVM) to the classification of sequences [7, 8]. An SVM is trained by minimising the empirical loss over the set of examples instead of the likelihood function, typically leading to superior run-time performance. However, training requires the choice of a suitable loss function. Common loss functions include the misclassification rate, the (negative) precision at a fixed value of recall, the $F_1$ loss and several others [9]. Depending on the application, certain loss functions prove more suited than others to reflect a desirable performance. For instance, in the case of sparse detection, a balanced combination of precision and recall such as the $F_1$ loss is certainly more meaningful than the simple misclassification rate. For this reason, Joachims in [10] has provided algorithms for training an SVM under several, diverse loss functions. However, they do not cover the targeted case of sequential labeling. Therefore, in this paper we present an algorithm for training a sequential SVM using the $F_1$ loss as loss function. The experimental results over a challenging sequential action dataset (the TUM Kitchen Dataset depicting human actions in a kitchen setting [11]) show that the proposed method achieves higher performance than the state of the art in terms of both $F_1$ measure and classification accuracy.

2. SEQUENTIAL LABELING AND STRUCTURAL SVM

2.1. Sequential labeling

The formal goal of sequential labeling is to predict a sequence of class labels, $y = (y_1, \ldots, y_T)$, for a given measurement sequence, $x = (x_1, \ldots, x_t, \ldots, x_T)$, where $x_t$ is a feature vector at sequence position $t$ and $y_t$ is a corresponding discrete label, $y_t \in 1 \ldots M$. In many cases, it is not restrictive to assume that $y_t$ is a binary label (1: positive class; 0: negative class), obtaining multi-class classification as a combination of binary classifiers. Therefore, in the following we focus on this binary case. The most widespread model for sequential labeling is the hidden Markov model which is a probabilistic graphical model factorising the joint probabilities of labels and measurements as:
\[ p(y, x) = p(y_1) \prod_{t=2}^{T} p(y_t | y_{t-1}) \prod_{t=1}^{T} p(x_t | y_t) \]  

The terms in the first product, \( p(y_t | y_{t-1}) \), are known as transition probabilities and the terms in the latter, \( p(x_t | y_t) \), are known as emission densities. By restricting the emission densities to belong to the exponential family of distributions, i.e. \( p(x_t | y_t) \propto \exp(w_{em}^T f(x_t, y_t)) \), and moving to a logarithmic scale, it is easy to see that the HMM joint log-probability is equivalent to the score a generalised linear system:

\[
\ln p(y, x) \propto w^T \phi(y, x) = w_{init}^T f(y_1) + \\
+ \sum_{t=2}^{T} w_{tran}^T f(y_t, y_{t-1}) + \sum_{t=1}^{T} w_{em}^T f(x_t, y_t) 
\]

This generalised linear model is more suitable for discriminative training and notable cases are conditional random fields [12] and the structural SVM [8]. The inference problem consists of determining the best class sequence for a given measurement sequence:

\[
\bar{y} = \arg \max_y p(y, x) = \arg \max_y (w^T \phi(y, x)) 
\]

This problem can be efficiently solved in \( O(T) \) time by the Viterbi algorithm working in logarithmic scale. Viterbi is an instance of dynamic programming where the sequence is scanned in a left-to-right manner and partial solutions accumulated. The final accumulated solution is guaranteed to be the global optimum [6].

2.2. Loss function

A loss function quantifies the loss, or cost, carried by a misclassification. If we assume that each sample carries a single class label, we can note the true label of sample \( i \) as \( y^i \), our prediction as \( y \), and the loss function as \( \Delta(y^i, y) \). The simplest loss function is the zero-one loss, assigning a unit loss to every misclassification and a zero loss to every correct classification:

\[
\Delta_{0-1}(y^i, y) = \begin{cases} 
1 & \text{for } y \neq y^i \\
0 & \text{for } y = y^i 
\end{cases} 
\]

In this case, the average loss over a sample set is just equivalent to the misclassification, or error, rate. However, other loss functions are required for differentiating between different types of error. This is particularly important in sequential labeling, where a sequence of labels is predicted at once: we certainly wish to differentiate between major errors, where many labels may be predicted wrongly, and minor errors with only a few, incorrect labels. In this case, the most common loss function is the Hamming loss that adds a loss fraction for every incorrect label [9]:

\[
\Delta_{Hamming}(y^i, y) = \frac{1}{T} \sum_{t=1}^{T} y^i_t \oplus y_t 
\]

where operator \( \oplus \) is the modulo-2 sum, or XOR boolean operator.

Although the Hamming loss is a more appropriate loss function than the zero-one loss for sequences as it accounts for gradual losses, it does not reflect desirable performance in some important cases. Let us consider a simple example: the ground truth, \( y^i \), having 3 positive labels and 97 negative labels. In this case, a trivial classifier with constant negative output would carry a loss of only 3%. This asymmetric situation is very common in detection, where positive labels are generally less frequent than negative ones. A useful performance figure that accounts specifically for false positive and false negative errors is the \( F_1 \) measure:

\[
F_1(y^i, y) = \frac{2 \cdot TP}{2 \cdot TP + FN + FP} 
\]

where \( TP \) is the number of correctly classified positive samples, \( FN \) the number of false negatives and \( FP \) the number of false positives. The \( F_1 \) measure is a figure of merit ranging between 0 (worst performance) and 1 (best) and it is then straightforward to define a loss function based on the \( F_1 \) measure as:

\[
\Delta_{F_1}(y^i, y) = 1 - F_1(y^i, y) 
\]

In the case of the trivial classifier with constant negative output, the \( F_1 \) loss reaches its maximum value (100%) since \( TP \) is equal to zero. Other important loss functions are based on: a) the precision \( (TP/(TP + FP)) \) at a fixed value of recall \( (TP/(TP + FN)) \), b) the recall at a fixed value of precision, c) precision and recall at equal value, d) the \( F_\beta \) measure (a variant of the \( F_1 \) measure weighing \( FP \) and \( FN \) differently). All such losses can be similarly computed from the values of \( TP, FP, FN \) (the contingency table of prediction \( y \) versus ground truth \( y^i \)).

2.3. Structural SVM

SVM has been used widely in classification problems mainly thanks to its remarkable experimental accuracy [13]. The conventional SVM is a classifier for independent data, i.e. data where the \{label, measurement\} pairs are assumed independent of each other. However, the SVM has also been extended to the prediction of structured labels, i.e. multiple labels that have mutual dependencies in the form of sequences, trees and graphs and that possibly co-depend on multiple measurements [7, 8]. Given a set of \( N \) training instances \{\( y^i, x^i \), \( i = 1 \ldots N \)}, structural (or structured-output) SVM finds the optimal model’s parameters by constrained minimisation of the following objective function:
\[
\sum_{i=1}^{N} \xi^i + \frac{1}{C} \|w\|^2 \quad \text{s.t.} \\
w^T \phi(y^i, x^i) - w^T \phi(y, x^i) \geq \Delta(y^i, y) - \xi^i, \\
\xi^i \geq 0, i = 1 \ldots N, \forall y
\] (8)

The first term in the objective function (\(\sum_{i=1}^{N} \xi^i\)) can be proven a convex upper bound for the loss over the training set, showing that the SVM is a minimum-empirical-loss classifier. The second term (\(\|w\|^2\)) is a regulariser that disambiguates the scale of the \(w\) parameters. Parameter \(C\) is an arbitrary, positive coefficient that balances the two terms in the objective. In the constraints, function \(w^T \phi(y, x)\) assigns a generalised linear score to a \(\{y, x\}\) pair. In the case of sequential labeling, it is given by Eq. (2). Eventually, \(\Delta(y^i, y)\) is the chosen loss function.

Due to the exponential number of values for \(y\) in Eq. (8), exhaustive verification of the constraints is infeasible. However, [8] has shown that it is possible to find \(c\)-correct solutions in polynomial time by using only the “most violated” constraint for each sample, i.e., the class labels with the highest sum of score and loss:

\[
y = \arg\max_y (w^T \phi(y, x) + \Delta(y^i, y))
\] (9)

This problem is commonly referred to as “augmented inference”. In the case of the zero-one loss or the Hamming loss, the augmented inference for sequences can be easily resolved by an appropriately weighted Viterbi algorithm. However, when the required loss is the \(F_1\), or any other non-decomposable loss functions, the Viterbi algorithm is not applicable anymore. An algorithm for these cases is the main contribution of this paper and is presented in the following section.

3. AUGMENTED INFERENCE FOR BINARY SEQUENTIAL LABELING UNDER THE \(F_1\) LOSS

The solution of Eq. (9) is the crux of structural SVM for sequential labeling. Since the \(F_1\) loss is not additive over the sequence’s frames, substantial changes to conventional dynamic programming algorithms are required. We start by re-writing Eq. (7) as:

\[
\Delta_{F_1}(y^i, y) = 1 - \frac{2(P - FN)}{2P + FP - FN}
\] (10)

Given any arbitrary ground-truth label sequence, \(y^i\), the number of positive labels, \(P\), in Eq. (10) is fixed. Therefore, the value of the \(F_1\) loss only further depends on the number of false positives, \(FP\), and false negatives, \(FN\), in the predicted sequence, \(y\). We present our strategy herewith: by making the prediction in left-to-right order along the sequence, the values of \(FP\) and \(FN\) can only increment or remain unchanged.

We can thus still approach the solution of Eq. (9) by dynamic programming by extending the state of a partial solution to include: a) the label of the current class, \(y_t\), as in conventional Viterbi; b) the number of false positives, \(FP\), in sub-sequence \(y_{1:t}\); and c) the number of false negatives, \(FN\), in sub-sequence \(y_{1:t}\). We use notation \(\psi(FP, FN, y_t)\) to indicate the \(y_{1:t}\) sub-sequence with the highest score for the given extended state, and we note its score as \(p(\psi)\). The generic induction step is as follows: at any time step, \(t\), a partial solution is obtained by extending two of the partial solutions of time \(t - 1\) with the current prediction, \(y_t\), and the corresponding increment of either \(FP\) or \(FN\) if the prediction is incorrect, or neither if correct. After the final time step, \(T\), Eq. (9) is computed over all the stored sequences and the argmax returned.

Algorithm 1 describes the solution formally.

During the computation of Algorithm 1, the number of stored sequences increases at every time step and reaches its maximum at the last time step, \(T\). Given that \(FP\) and \(FN\) are bound by the number of negative and positive labels, \(N\) and \(P\), respectively (with \(N + P = T\)), the number of stored sequences is upper bound by \(2(P + 1)(N + 1)\). The final loop of Algorithm 1 runs over all values of \(FP, FN\) to find the maximum for Eq. (9): with obvious modifications, the final loop can be adapted to search for the maximum over fixed values of \(FP\) or \(FN\), at equal rates \(FP = FN\) or, in general, any loss functions based on the contingency table.

4. EXPERIMENTAL RESULTS

The recognition of human activities in video is a major application of sequential labeling [14]. In this section, we evaluate our method on the TUM Kitchen Dataset which is a collection of activity sequences recorded in a kitchen environment equipped with multiple sensors [11]. In the kitchen environment, various human subjects were asked to set a table in different ways, performing actions such as Reaching, Carrying, TakingSomething and several others. The dataset contains multiple, simultaneous types of data: video data from 4 fixed overhead cameras, motion capture data extracted from the videos, RFID tag readings from 3 fixed readers embedded in the environment, magnetic sensors readings from objects and the environment, and their annotated action labels. For our experiments, we have used the motion capture data (3-D body joint locations) which include 19 sequences for the left and right hands separately, each ranging in length between 1,000 and 6,000 frames. Each measurement is a 45-D vector.

We developed our software in MATLAB 2011 using the original SVMstruct from Thorsten Joachims and its MATLAB wrapper from Andrea Vedaldi [15, 16]. All experiments were performed on a PC with an Intel i7 2.4GHz CPU with 8 GB RAM. As data set, we divided the 19 left hand sequences into a training set with 6 sequences and a test set with the remaining 13, and we considered all the actions that had pos-
Table 1: Comparison of the average accuracy and average $F_1$ measure on TUM kitchen dataset with different loss functions.

<table>
<thead>
<tr>
<th>Loss function</th>
<th>Accuracy (%)</th>
<th>$F_1$ measure (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reaching</td>
<td>20.02</td>
<td>61.39</td>
</tr>
<tr>
<td>TakingSomething</td>
<td>21.27</td>
<td>61.59</td>
</tr>
<tr>
<td>LoweringAnObject</td>
<td>48.51</td>
<td>52.90</td>
</tr>
<tr>
<td>ReleasingGrasp</td>
<td>22.91</td>
<td>60.30</td>
</tr>
<tr>
<td>Carrying</td>
<td>40.89</td>
<td>53.36</td>
</tr>
<tr>
<td>Average</td>
<td>30.72</td>
<td>52.96</td>
</tr>
</tbody>
</table>

Algorithm 1: Algorithm for computing argmax in Eq. (9)

Input: model $w$, measurements $x = (x_1, \ldots, x_T)$, ground-truth classes $y^g = (y_1^g, \ldots, y_T^g)$

Output: predicted classes $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_T)$

Initialize: $FP_{max} = FN_{max} = 0$

// NB: [ ]: string concatenation operator,
// $\psi$(invalidarg) = NULL, $p$(NULL) = $-\infty$

for $t = 1 : T$

if $y_t^g = 0$ then 

$FP_{max} = FP_{max} + 1$

else 

$FN_{max} = FN_{max} + 1$

for $FP = 0 : FP_{max}$

for $FN = 0 : FN_{max}$

if $y_t^g = 0$ then 

$\psi(FP, FN, y_t) = \arg\max\{p(\psi(FP, FN, y_t - 1) = 0), 0)\}$

$p(\psi(FP, FN, y_t = 1) = 0)\}

else 

$\psi(FP, FN, y_t = 1) = \arg\max\{p(\psi(FP - 1, FN, y_t - 1) = 0), 1)\}$

$p(\psi(FP - 1, FN, y_t = 1) = 1)\}$

for $FP, FN, y_t = 0 : FP_{max}$

if $temp > best$ then 

$\hat{y} = \arg\max\{p(\psi(FP, FN, y_t = 0)\}$

$p(\psi(FP, FN, y_t = 1)\}$

return $\hat{y}$

5. CONCLUSION AND FUTURE WORK

In this paper, we have proposed an algorithm for training structural SVM for sequential labeling under an $F_1$ loss. Such a loss is a more appropriate performance measure than the frame-level accuracy when a balance between precision and recall is sought and is particularly useful in detection scenarios. The proposed algorithm leverages the notion of a state extended with accumulated errors and retains the efficiency of dynamic programming. Experiments conducted on the challenging TUM Kitchen Dataset have reported an improvement of 8.9 percentage points in $F_1$ measure and 4.3 percentage points in frame-level accuracy over the runner-up (Hamming loss training). In addition, the proposed algorithm can be used for any other loss functions similarly derived from the contingency table of the predicted sequence versus the ground truth. In the near future, we plan on extending the proposed approach to the case of joint multi-label sequential labeling.
6. REFERENCES


