Partial Sequence Labeling With Structured Gaussian Processes
Xiaolei Lu and Tommy W. S. Chow, Fellow, IEEE

Abstract—Existing partial sequence labeling models mainly focus on a max-margin framework that fails to provide an uncertainty estimation of the prediction. Furthermore, the unique ground-truth disambiguation strategy employed by these models may include wrong label information for parameter learning. In this article, we propose structured Gaussian processes for partial sequence labeling (SGPPSL), which encodes uncertainty in the prediction and does not need extra effort for model selection and hyperparameter learning. The model employs factor-as-piece approximation that divides the linear-chain graph structure into the set of pieces, which preserves the basic Markov random field structure and effectively avoids handling a large number of candidate output sequences generated by partially annotated data. Then, confidence measure is introduced in the model to address different contributions of candidate labels, which enables the ground-truth label information to be utilized in parameter learning. Based on the derived lower bound of the variational lower bound of the proposed model, variational parameters and confidence measures are estimated in the framework of alternating optimization. Moreover, a weighted Viterbi algorithm is proposed to incorporate confidence measures to sequence prediction, which considers label ambiguity arose from multiple annotations in the training data and thus helps improve the performance. SGPPSL is evaluated on several sequence labeling tasks and the experimental results show the effectiveness of the proposed model.

Index Terms—Partial sequence labeling, structured Gaussian processes (GPs), variational lower bound, weighted Viterbi.

I. INTRODUCTION

SEQUENCE labeling, which refers to assign a label to each token in a given input sequence, has been successfully applied in natural language processing (NLP) and computational biology. For example, as shown in Fig. 1, in part-of-speech (POS) tagging, sequence labeling assigns the POS tag to each word in a given sentence. Fig. 1 also shows how sequence labeling detects the mention of genes from biomedical publication abstracts. In these tasks, label assignment for each element should consider the surrounding context of this element in the sequence (e.g., the preceding element and its corresponding label). Sequence labeling can help explore the structure of the given contexts and provide globally optimal label sequence for the input sequence.

Manuscript received 19 July 2020; revised 24 May 2021 and 21 November 2021; accepted 12 July 2022. Date of publication 25 July 2022; date of current version 6 February 2024. This work was supported by the Hong Kong Government GRF under Project CityU 11216620. (Corresponding author: Tommy W. S. Chow.)

The authors are with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong (e-mail: xiaoleilu2-c@my.cityu.edu.hk; eetchow@cityu.edu.hk).

Digital Object Identifier 10.1109/TNNLS.2022.319726

Many effective methods, such as structured support vector machine (S-SVM) and conditional random fields (CRFs), have delivered promising results to sequence labeling. A linear-chain CRF [1] directly models the conditional probability of the label sequence without assumption on the dependencies among the observations. They achieve good performance in the tasks of POS tagging and named entity recognition (NER). However, these traditional methods heavily rely on task-specific feature engineering. During the past few years, a variety of deep sequence labeling models have been proposed to improve the performance with good feature representation learning. For example, bidirectional long-short term memory (Bi-LSTM) [2] is designed to access past features and future features for a specific token in the sequence. Bi-LSTM with CRF on top layer (Bi-LSTM-CRF) [3] effectively makes use of the label information from past and future tokens to predict the tag of current token. However, the above models usually require a large amount of training data with complete annotations [4]–[6], which is costly and laborious to produce. Although semisupervised sequence labeling models [7] can utilize unlabeled data to facilitate learning from a small amount of labeled data, they still need exact annotations.

In reality, it is more cost-effective to obtain a set of candidate labels for the instance [8]. For example, well-developed commercial crowdsourcing platforms [e.g., Amazon Mechanical Turk (AMT)] provide a cheap and efficient way to obtain large labeled data, where the labeling task is divided into many subtasks that are distributed to a large group of ordinary workers. However, the quality of these crowd annotations cannot be guaranteed as the expertise level of workers varies. Fig. 2 shows a sentence with crowded named entity labelings on AMT, where each word in the sentence is ambiguously annotated and only annotator 3 provides correct labelings. Therefore, utilizing ambiguous annotations to train a prediction model with high performance is practically significant for sequence labeling.

Recently, partial label learning (PLL) has been introduced to sequence labeling [8], [9]. In partial sequence labeling (PSL),
the ground-truth label is masked by ambiguous annotations. As shown in Fig. 3, each word in a given sentence is annotated with ambiguous POS tags where both the ground-truth and noisy labelings are included. Compared with golden single annotations, the ambiguously annotated sentence is of poor quality as it additionally provides $3^{10} - 1$ noisy labelings, which adversely affects the prediction performance of standard supervised learning models. PSL-based methods mainly focus on identifying the ground-truth label from ambiguous annotations. For example, Lou and Hamprecht [9] extended convex learning from partial labels to sequence labeling that aims to discriminate the ground-truth output sequence from other possible outputs.

Existing partial sequence labeling models are based on a max-margin framework [8], [9]. While these models are able to deliver good prediction performance by learning from partial annotations, they only produce deterministic outputs and do not quantify the uncertainty in prediction that shows how confident we can be in interpreting the results. In partial sequence labeling, due to inherent model uncertainty [10] and noisy input, it is important to measure how confident the PSL model is about prediction. Furthermore, cross validation for model selection and hyperparameter learning in max-margin-based PSL models may be computationally infeasible as the search space is too large. For example, there are $n^2$ combinations of two regularization parameters in two large margin formations [8] for partial sequence labeling, where $n$ is the number of possible values for each regularization parameter.

In this article, to address the above problems in partial sequence labeling, we propose structured Gaussian processes for partial sequence labeling (SGPPSL) that estimates the posterior distribution of model parameters and posterior predictive distribution, which encodes uncertainty in the prediction and does not need cross validation for model selection and hyperparameter learning. By investigating the combination of Gaussian processes (GPs) and CRFs in modeling sequential data, we develop an effective nonparametric Bayesian model to learn from partially annotated sequential data. Our contributions can be summarized as follows.

First, we develop a structured GP piecewise-likelihood model. The model employs the factor-as-piece approximation that divides the linear-chain graph structure into the set of pieces, which preserves the basic Markov random field structure and effectively avoids handling a large number of candidate output sequences generated by partially annotated data. Furthermore, different from unique disambiguation strategy, confidence measure is introduced to address the different contributions of candidate labels, which enables the ground-truth label information to be utilized in parameter learning.

Second, based on the obtained variational lower bound $L$ of the proposed model, we further derive the lower bound of $L$, which aims to solve the nondifferentiable problem in optimization. Variational parameters and confidence measure are estimated in the framework of alternating optimization.

Third, the weighted Viterbi algorithm is proposed to include estimated confidence measures in sequence prediction. For a given test instance, by compressing the confidence measures collected from its nearest neighbors in the training data into a confidence factor and then incorporating it to the score computation in decoding, label ambiguity arose from multiple annotations can be considered in state emission and transition.

Fourth, we evaluate the proposed method on three NLP tasks: base NP, chunking, and NER. The experimental results show that in most cases, our proposed model outperforms the baselines.

The rest of this article is organized as follows. Section II reviews the related work. We describe the formulation, inference, and prediction of the proposed SGPPSL in Section III. A series of experiments is presented in Section IV. Section V summarizes this article and discusses the possible future research.

II. RELATED WORK

Hidden Markov models (HMMs) [11], [12] and CRFs [13], [14] are the most widely used graph models for sequence labeling. Many variants of CRFs like semi-Markov conditional random fields (semi-CRFs) [15] and hierarchical conditional random fields (HCRFs) have been proposed for modeling complex structured outputs. Furthermore, in recent years, deep sequence labeling models that combine deep learning and graph models, such as CRF-CNN [16] and LSTM-CRF [17], have achieved competitive results compared with traditional graph models.

Since supervised-based models require a large number of training data with exact annotations, semisupervised sequence labeling has been investigated to effectively utilize a large number of unlabeled data for training, which greatly reduces labor costs and improves the efficiency of data collection. For example, semisupervised CRFs [7] are proposed to minimize the conditional entropy on unlabeled training instances. Brefeld and Scheffer [18] developed co-training principle into support vector machine to minimize the number of errors for labeled data and the disagreement for the unlabeled data. While these semisupervised sequence labeling models partly reduce the demand for large labeled datasets, exact annotations are essential for parameter learning.

PLL gets increasing attention to handle one of the challenging classification problems where the true label is masked by multiple ambiguous annotations. There has been much research focusing on different disambiguation strategies. The most intuitive way is to treat each candidate label equally and average the scores of all candidate labels. For example, Cour et al. [19] proposed convex loss for partial labels (CLPL) to disambiguate candidate labels with non-candidate labels. However, the average disambiguation (AD) strategy tends to incorporate wrong label information. To identify the true label from candidates, unique disambiguation
strategy has been widely accepted. Partial label support vector machine (PL-SVM) [20] was formulated to maximize the margin between the best wrong prediction of noncandidate labels and the current prediction of the ground-truth label. Yu and Zhang [21] proposed a maximum margin PLL that aims to address the difference between the ground truth and other candidates. “EM + Prior” model [22] generalizes the EM model with prior knowledge indicating which label is more likely to be the ground truth and then estimates the ground truth in iterative optimization. Moreover, Zhang et al. [23] used an error-correcting output code (ECOC) coding matrix to treat the candidate label set as an entirety (PL-ECOC), which avoids the disambiguation between candidate labels. While PLL fails to address some candidates that could be false positives or similar to the ground-truth label. As shown in Fig. 3, “following” can be identified as “JJ” (i.e., adjective) by unique disambiguation, which may adversely affect the learning of prediction model.

III. PROPOSED MODEL

A. Model Formulation

Given sequence data \( \{X_i, Y_i\}_{i=1}^N \), \( X^i = \{x_1, \ldots, x_m\} \), and \( Y^i = \{y_1, \ldots, y_m\} \), \( y_m = \{y_1, \ldots, y_l\} \), where \( l \) denotes the number of candidate labels. \( Y \) is the label set for \( y \). If \( m \)th element in the \( i \)th sequence is correctly annotated, then \( y_m \) is a singleton. First, consider the factor graph of linear-chain CRF, as shown in Fig. 4, where unary factor \( \psi(y_i, x_i) \) and transition factor \( \psi(y_i, y_{i-1}, x_i) \) are included. For the partially annotated sequence \( X_i \), there are \( l^m \) possible candidate output sequences. To avoid handling a large number of candidates, we employ piecewise training and use factor-as-piece approximation [30] to divide the full graph into pieces, where each factor is assigned to its own piece. Then, the number of candidate outputs can be reduced to \( lm + l^2(m-1) \).

Based on the divided factor graph, the likelihood of \( p(Y_i|X_i) \) is defined as

\[
p(Y_i|X_i) = \prod_{a=1}^M \prod_{y_a \in S_a} C_a^j p(y_a^j|x_a)
\]

(1)

where \( M \) is the number of factors in the linear-chain structure of \( \{X_i, Y_i\} \), \( S_a \) denotes the set of candidate labels for \( a \)th factor. \( C_a^j \) represents the confidence measure for the \( j \)th candidate label in \( S_a \). The conditional distribution of \( p(y_a|x_a) \) is defined as

\[
p(y_a|x_a) = \begin{cases} \frac{\psi(y_a, x_a)}{\sum_{y_{a-1}} \psi(y_{a-1}, x_{a-1}),} & \text{unary factor} \\ \frac{\psi(y_a, y_{a-1}, x_{a-1})}{\sum_{y_{a-2}} \psi(y_{a-2}, y_{a-1}, x_{a-1})}, & \text{transition factor} \end{cases}
\]

(2)

As shown in (1), the modeling structured output for each factor is weighted by \( C_a^j \), which addresses the different contributions of candidate labels. Compared with unique ground-truth identification strategy, confidence-weighted mechanism enables the ground-truth label to be utilized in the learning process.
GP classification introduces latent variables (LVs) to mediate the influence of the input and the output. We define LV per factor. There are two types of LVs corresponding to different factor types in the linear-chain structure: the unary LV \( f_a \) and the transition LV \( f_t \), \( f_a(y) = \{ f_a(x_1, y), \ldots, f_a(x_{N \times m \times l, y}) \} \), where \( f_a(x, y) \) is associated with a specific label \( y \) for each training instance \( x \). The transition LV \( f_t \) is defined over all label pairs \((y, y')\), where \( y, y' \in \mathcal{Y} \). Based on the defined LVs, given partially annotated sequence data \( \{X_i, Y_i\}_{i=1}^N \), the likelihood of \( p(Y|X, f) \) is defined as

\[
p(Y|X, f) = \prod_{i=1}^N \prod_{y_i, y_{i+1} \in \mathcal{S}_{a}} C_{ia} \exp \left( \frac{f(x_{ia}, y_{ia})}{\sum_{y'} \exp(f(x_{ia}, y'))} \right) \tag{3}
\]

where \( f = \{ f_U, f_T \} \). \( f_U \) is the collection of \( f_t \). The number of \( f_t \) is fixed with \( \| \mathcal{Y} \|^2 \), while the number of \( f_a \) grows with the size of data. \( f_U \) is \( f_T \) defined over all possible labels in \( \mathcal{Y} \). There are total \( N \times m \times l \times \| \mathcal{Y} \| \) unary LVs in \( f_U \).

**B. GP Prior and Posterior**

The LVs \( f_U(y) \) is drawn from a zero-mean GP with covariance function \( \mathcal{K}_u(y) \) of size \( Nml \times Nml \), where \( \mathcal{K}_u(i, j) \) is defined as

\[
k(x_i, x_j) = \exp\left(-\theta_k \| x_i - x_j \|^2 \right) \tag{4}
\]

where \( \theta_k \) is a kernel hyperparameter.

The transition LVs \( f_T \) is defined with a zero mean and covariance \( \mathcal{K}_T \) and \( \mathcal{K}_T(i, j) \) is a covariance function of label pairs \((y_i, y'_j)\) and \((y'_j, y'_i)\), which takes the following form:

\[
\left\langle \left( y_i = y'_j \wedge y'_j = y'_i \right) \right\rangle. \tag{5}
\]

Based on the above specification for \( f_U \) and \( f_T \), the GP prior over \( f \) is defined as

\[
p(f|X) = N(f; 0, K) = N\left( \begin{bmatrix} f_U \; f_T \end{bmatrix}; 0, \begin{bmatrix} K_u & 0 \\ 0 & K_T \end{bmatrix} \right) \tag{6}
\]

where \( K_U \) is block-diagonal with \( \| \mathcal{Y} \| \) equal size blocks and \( K_T = I_{\| \mathcal{Y} \|} \).

The posterior distribution over \( f \) is

\[
p(f|X, Y) = \frac{p(Y|X, f)p(f|X)}{p(Y|X)} \tag{7}
\]

where

\[
p(Y|X) = \int p(Y|X, f)p(f|X)df. \tag{8}
\]

**C. Variational Gaussian Approximate Inference**

The posterior distribution \( p(f|X, Y) \) cannot be calculated analytically as the non-Gaussian property of \( p(Y|X, f) \). The most common way is to approximate the posterior with a tractable Gaussian distribution.

First, variational inference approximates \( p(f|X, Y) \) with a variational distribution \( q(f) \) by using the following criterion:

\[
\min_{q(f)} KL(q(f)||p(f|X, Y)) \tag{9}
\]

where \( KL \) refers to the Kullback–Leibler divergence.

The log likelihood of \( p(Y|X) \) can be derived as follows:

\[
\log p(Y|X) = \int q(f) \log p(Y|X, f)df = \int q(f) \log \frac{p(Y|X, f)p(f|X)}{p(f|X, Y)q(f)}df = \int q(f) \log \frac{q(f)}{p(f|X, Y)}df + \int q(f) \log \frac{p(Y|X, f)p(f|X)}{q(f)}df = KL(q(f)||p(f|X, Y)) + L(q(f)) \tag{10}
\]

where \( L(q(f)) \) is the evidence lower bound (ELBO).

We have \( \log p(Y|X) = L(q(f)) \) due to the nonnegative property of \( KL(q(f)||p(f|X, Y)) \). Then, the final optimization of variational inference is to maximize \( L(q(f)) \), which can be defined as

\[
L(q(f)) = \int q(f) \log \frac{p(f|X, Y)p(Y|X, f)}{q(f)}df = -KL(q(f)||p(f|X, Y)) + \int q(f) \log p(Y|X, f)df = -KL(q(f)||p(f|X, Y)) + \sum_{i=1}^N \sum_{a=1}^l \sum_{y_j \in \mathcal{S}_{ia}} \left[ \log p(y_{ia}'|x_{ia}, f) + \log(C_{ia}) \right]. \tag{11}
\]

Variational Gaussian approximate inference [31] assumes the posterior \( q(f) \) to be a multivariate Gaussian

\[
q(f) = N(f; \mu, V) = N\left( \begin{bmatrix} f_U \; f_T \end{bmatrix}; \begin{bmatrix} \mu_U, & \mu_T \end{bmatrix}, \begin{bmatrix} V_U & 0 \\ 0 & V_T \end{bmatrix} \right). \tag{12}
\]

Based on the closed-form expression of the KL divergence between two Gaussians, \( KL(q(f)||p(f|X, Y)) \) can be written as

\[
KL(q(f)||p(f|X, Y)) = \frac{1}{2} \left[ \log |V^{-1}K| + \text{tr}(V^{-1}K^{-1})d + \mu^T K^{-1} \mu \right] \tag{13}
\]

where \( d \) is a constant and equals the number of variational parameters.

The expectation \( \mathbb{E}_{q(f)}[\log p(y_{ia}'|x_{ia}, f)] \) is tractable as \( p(y_{ia}'|x_{ia}, f) \) is a softmax function. We employ Jensen’s inequality to obtain the tractable lower bound of \( \mathbb{E}_{q(f)}[\log p(y_{ia}'|x_{ia}, f)] \), which is stated as follows:

\[
\mathbb{E}_{q(f)}[\log p(y_{ia}'|x_{ia}, f)] = \mathbb{E}_{q(f)} \left[ \log \frac{\exp(f(x_{ia}, y_{ia}'))}{\sum_{y'} \exp(f(x_{ia}, y'))} \right] = \mathbb{E}_{q(f)} \left[ f(x_{ia}, y_{ia}') \right] - \mathbb{E}_{q(f)} \left[ \log \sum_{y'} \exp(f(x_{ia}, y')) \right] \geq \mathbb{E}_{q(f)} \left[ f(x_{ia}, y_{ia}') \right] - \log \sum_{y'} \mathbb{E}_{q(f)}[\exp(f(x_{ia}, y'))] = \mu(x_{ia}, y_{ia}') - \log \sum_{y'} \exp[\mu(x_{ia}, y') + \frac{1}{2} V_{(x_{ia}', y_{ia}')}}] \tag{14}
\]
where \( x_i^j \) refers to the \( i \)th factor tagged with the \( j \)th candidate label.

Then, the optimization problem turns into maximizing the lower bound of \( L(q(f)) \), which is defined as

\[
L(q(f)) = \frac{1}{2} \log |VK^{-1}| - \text{tr}(VK^{-1}) + d - \mu^T K^{-1} \mu
\]

\[
+ \sum_{i=1}^{N} \sum_{a=1}^{M_i} \sum_{y_i^j \in S_a} \mu(x_{ia}, y_i^j) - \log \sum_{y} \exp \left[ \mu(x_{ia}, y^j) + \frac{1}{2} V((x_{ia}', y^j), (x_{ia}', y^j)) \right] \]

\[
+ \sum_{i=1}^{N} \sum_{a=1}^{M_i} \sum_{y_i^j \in S_a} \log (C_{ia})
\]

where for the formula taking the form, \( \mu(x_i, y_i) - \log \sum_y \exp \mu(x_i, y) + (1/2)V((x, y), (x, y)) \) is computed with the following:

**Unary Factor:**
\[
\mu_U(y) = \log \sum_y \exp \left[ \mu_U(y') + \frac{1}{2} V_{U(y)}(y') \right].
\]

**Transition Factor:**
\[
\mu_T(y_{i-1}, y_i) = \log \sum_y \exp \left[ \mu_T(y_{i-1}, y_i) + \frac{1}{2} V_T((y_{i-1}, y'), (y_{i-1}, y')) \right].
\]

**D. Parameter Estimation**

The parameters \( \{(\mu, V), \theta, C\} \) are required to be estimated, where \( \theta \) is the set of \( \{\theta_i\} \) for \( K \) and \( C = \{c_d^j\} \) denotes the set of confidence measure of each candidate label for a training instance. Alternating optimization is employed to estimate parameters.

1. **Fixed \( \{(\mu, V), \theta\} \):** Each element \( c_d^j \) in \( C \) is computed with

\[
c_d^j = \frac{\exp \left( \mu(x_{d}, y_i^j) \right) + \frac{1}{2} V((x_{d}, y_i^j), (x_{d}, y_i^j))}{\sum_j \exp \left( \mu(x_{d}, y_i^j) \right) + \frac{1}{2} V((x_{d}, y_i^j), (x_{d}, y_i^j))}.
\]

\( p(y_{i|d}|x_{d}) \) takes the form of \( \int p(y_{i|d}|x_{d}, f)p(f)df \) where the computation of expectation of softmax function is intractable. We employ the strategy [29] that computing softmax of the expectation for LVs to obtain the probability of \( p(y_{i|d}|x_{d}) \) and then compute \( c_d^j \) as shown in (17).

2. **Fixed \( C \):** Optimizing \( \{(\mu, V), \theta\} \) by solving the maximization of \( L_q(f) \). Based on the concavity of \( L_q(f) \), parameter optimization can be realized by the nested loop. In the inner loop, we use a coordinate ascent algorithm to estimate \( \{(\mu, V) \} \) with fixed \( \theta \). Then, the hyperparameter \( \theta \) is estimated in the outer loop for the fixed \((\mu, V)\). The gradients with respect to \( \{(\mu, V), \theta\} \) is computed as follows:

\[
\nabla_\mu L_i = -K^{-1} \mu + \frac{\partial P}{\partial \mu_{(i, a, j)}}
\]

\[
\nabla_V L_i = \frac{1}{2} (V^T - K^{-1}) + \frac{\partial P}{\partial V_{(i, a, j)}}
\]

\[
\nabla_\theta L_i = \left( K^{-1} \mu \right)(\partial K / \partial \theta)(K^{-1})^T + \text{tr} \left[ K^{-1} (V K^{-1} - I) (\partial K / \partial \theta) \right]
\]

where \( t \) refers to the \( t \)th block as there are total \( |Y| + 1 \) blocks in \( V \) and

\[
P = \sum_{i=1}^{N} \sum_{a=1}^{M_i} \sum_{y_i^j \in S_a} \mu(x_{ia}, y_i^j)
\]

\[
- \log \sum_y \exp \left[ \mu(x_i^j, y^j) + \frac{1}{2} V((x_i^j, y^j), (x_i^j, y^j)) \right].
\]

E. Prediction

Given the estimated parameters \( \{(\mu, V), \theta, C\} \) and a new sequence data \( [x_s, y_s] \), the posterior distribution \( p(f_i|X, Y, x_s) \) is defined as

\[
p(f_i|X, Y, x_s) = \int p(f_i|X, x_s, f)p(f|X, Y)df
\]

\[
= N \begin{bmatrix} \mu_{sU} \\ \mu_{sT} \end{bmatrix} \begin{bmatrix} V_{sU} & 0 \\ 0 & V_{sT} \end{bmatrix}
\]

where \( \mu_{sT} = \mu_T \) and \( V_{sT} = V_T \). \( \mu_{sU} \) and \( V_{sU} \) are derived as

\[
\mu_{sU} = K_{sU} K_{sU}^{-1} \mu_U,
\]

\[
V_{sU} = K_{sU} - K_{sU} K_{sU}^{-1} V_{sU} K_{sU}^{-1} K_{sU}.
\]

The predictive probability of \( p(y_{si}|x_{si}, X, Y) \) is defined as

\[
p(y_{si}|x_{si}, X, Y) = \int p(y_{si}|f_i) p(f_i|X, Y, x_s)df_i
\]

where \( p(y_{si}|f_i) \) is defined as

\[
p(y_{si}|f_i) = \frac{\exp(f_i(y_{si}))}{\sum_{y} \exp(f_i(y'))}.
\]

Due to the intractability of the expectation of softmax function, we also use the same strategy stated in (17) to define the score for assigning \( y_{si} \) to \( x_{si} \) and transiting from \( y_{si(i-1)} \) to \( y_{si} \) as

\[
S(y_{si}|x_{si}) = \frac{\exp(\mu_{sU}(y_{si}) + \frac{1}{2} V_{sU}(y_{si}))}{\sum_{y} \exp(\mu_{sU}(y') + \frac{1}{2} V_{sU}(y'))}
\]

\[
S(x_{si(i-1)}|y_{si(i-1)}) = \frac{\exp(\mu_{sT}(y_{si(i-1)}, x_{si(i-1)}) + \frac{1}{2} V_{sT}(y_{si(i-1)}, y'))}{\sum_{y} \exp(\mu_{sT}(y_{si(i-1)}, y') + \frac{1}{2} V_{sT}(y_{si(i-1)}, y'))}.
\]

Based on the confidence measures collected from the K-nearest neighbors of \( x_{si} \) in the training data, we define
the confidence factor \( \tau_i(y_a) \) that denotes the confidence of assigning the label \( y_a \) to \( x_{si} \) as

\[
\tau_i(y_a) = \begin{cases} 
\frac{1}{K_i} \sum_{k=1}^{K_i} c_k(y_a), & y_a \in \mathcal{Y}_i \\
\frac{1}{K_i} \sum_{y \in \mathcal{Y} \setminus \mathcal{Y}_i} c_k(y), & y_a \in \mathcal{Y} \setminus \mathcal{Y}_i 
\end{cases}
\]

(29)

where \( \mathcal{Y}_i \) denotes the set of labels collected from the K-nearest neighbors of \( x_{si} \) in the training data. Since \( |\mathcal{Y}_i| \leq |\mathcal{Y}| \), for the assigned label that is not included in \( \mathcal{Y}_i \), we choose to average the summarization of all confidence measures obtained from nearest neighbors, which guarantees that the confidences of these labels are lower than that of the ground-truth label.

Then, the confidence factor of label transition \( \tau(y_1, y_2) \) is defined as \( (1/T) \sum_{y_1=1}^{T} c(y_1, y_2) \), where \( T \) denotes the number of transition factor taking the form \((y_1, y_2)\) in the training data. In most cases, partially annotated data cover diverse labels, and for the label transition that is not included in the training data, we set the corresponding confidence measure to 0 to avoid invalid label transition.

For the weighted Viterbi decoding, the score \( g_i(y_{(i-1)}, y_i) \) is defined as

\[
g_i(y_{(i-1)}, y_i) = \tau_i(y_{ai}) S(y_{ai}, x_{si}) + \tau(y_{(a)_{i-1}}, y_{ai}) S(y_{a(i-1)}, x_{si}).
\]

(30)

In Viterbi recursion, the optimal intermediate score \( \delta_i(s) \) for the \( r \)th token with label \( s \) is represented with

\[
\delta_i(s) = \max_{y_{i-1} \in \mathcal{Y}} [\delta_{i-1}(y_{i-1}) + g_i(y_{i-1}, s)].
\]

(31)

Finally, the optimal label sequence can be obtained by path backtracking.

IV. EXPERIMENTAL RESULTS

In this section, we perform experiments on three NLP tasks: base NP, chunking, and NER. These tasks aim to find the meaningful segments from input sequences, which greatly benefits most of the NLP applications, such as document summarization and question answering. However, in these tasks, exact annotations for collecting training data are not feasible because of words’ ambiguity. For example, annotators may be confused by “sprout” in identifying noun or verb phrase. Thus, applying partial sequence labeling can effectively handle ambiguous label annotations.

A. Datasets

Base NP: This task is to identify noun phrases for a given sequence, which can be used for many downstream tasks. For example, keyphrase extraction can be treated as assigning the identified noun phrase with the label “keyphrase” or “nonkeyphrase.” We use the dataset developed by Ramshaw and Marcus [32] in the NP chunking experiments.

Chunking: This task divides the sentence into smaller segments, which includes noun phrase, verb phrase, and prepositional phrase. We choose the chunking dataset from CoNLL-2000 shared task [33]. This dataset is collected from

Wall Street Journal (WSJ) sections, where sections 15–18 are training data (211,727 tokens) and section 20 is test data (47,377 tokens).

Named Entity Recognition: This task is to identify the entity types in the sentence. We choose the NER dataset from CoNLL-2003 shared task [34]. This dataset contains four entity types: location, organization, person, and miscellaneous.

Table I summarizes the details of three datasets. We further construct a relatively small size dataset for each task, which aims to avoid large matrix operation for sequential data. Furthermore, the setting of partial annotations is included, where \( cl \) refers to the number of candidate labels and \( p \) is the proportion of exactly annotated instances. For the generation of candidate labels, as there is no prior knowledge about the ambiguous annotations toward the target label, we conduct random sampling \( cl \) times and each randomly selects a negative label to the candidates.

B. Baselines

We compare the proposed SGPPSLS with the following baselines. These baselines include parametric and nonparametric models. Also, the disambiguation strategy employed in these models can be classified into four types: random disambiguation, AD, identification disambiguation (ID), and disambiguation-free. Representative approaches are stated as follows.

1) NAIVE [8]: Structured SVM model that randomly chooses a label from candidates adds the ground-truth label.
2) CLPL [9]: AD-based method that maximizes the margin between candidate labels and noncandidate labels.
3) K-Nearest Weighted Voting (WKNN) [24]: A nonparametric approach that assigns the nearest neighbors with different weights based on distance and treats candidate labels equally (AD type), suggested setup: \( k = 5 \).
4) PL-ECOC [23]: A disambiguation-free strategy by encoding partially labeled instances with ECOCs—suggested setup: the codeword length \( L = \lceil 10 \cdot \log_2(q) \rceil \).
5) PL-SVM [20]: The ID-based method that maximizes the margin between the ground-truth label and the best prediction of wrong label.
6) CLLP [8]: The ID-based method that incorporates two types of margins: the margin between the ground-truth label and other candidate labels; the margin between the ground-truth label and noncandidate labels.
7) PALOC [35]: The ID-based method that induces the multiclass classifiers with one-vs-one decomposition
where $y_i$ denotes the ground-truth label for each token in the sequence, C.

**C. Recovering the Ground-Truth Label**

To measure the generalization performance of the baselines and SGPPSL, we employ a fivefold cross-validation strategy to train these models on the selected datasets and report the average performance. Specifically, for SGPPSL cross-validation is set as an outer loop of the framework of parameter estimation as described in Section III-D.

### Table II

**Performance of Recovering the Ground-Truth Label (%)**

| Task     | $c_l = 0.1$ | $c_l = 0.3$ | $c_l = 0.5$ | $c_l = 0.7$ | $c_l = 0.9$ |
|----------|-------------|-------------|-------------|-------------|-------------|
| Base NP  | $72.04$     | $76.68$     | $84.06$     | $86.71$     | $89.13$     |
| $c_l = 3$ | $73.35$     | $76.63$     | $79.02$     | $80.39$     | $90.83$     |
| $c_l = 4$ | $75.30$     | $72.67$     | $75.56$     | $78.73$     | $84.17$     |
| Chunking | $70.85$     | $72.07$     | $72.42$     | $73.94$     | $75.79$     |
| $c_l = 3$ | $62.27$     | $64.70$     | $67.13$     | $67.54$     | $70.20$     |
| $c_l = 4$ | $60.50$     | $60.62$     | $62.87$     | $65.10$     | $65.97$     |
| NER      | $73.35$     | $73.77$     | $74.69$     | $84.43$     | $87.23$     |
| $c_l = 3$ | $70.88$     | $72.38$     | $72.06$     | $73.75$     | $77.61$     |
| $c_l = 4$ | $69.30$     | $69.05$     | $70.39$     | $73.77$     | $73.86$     |

| $c_l$ | $0.1$ | $0.3$ | $0.5$ | $0.7$ | $0.9$ |
|-------|-------|-------|-------|-------|-------|
| $c_l = 2$ | $72.04$ | $76.68$ | $84.06$ | $86.71$ | $89.13$ |
| $c_l = 3$ | $73.35$ | $76.63$ | $79.02$ | $80.39$ | $90.83$ |
| $c_l = 4$ | $75.30$ | $72.67$ | $75.56$ | $78.73$ | $84.17$ |
| $c_l = 2$ | $70.85$ | $72.07$ | $72.42$ | $73.94$ | $75.79$ |
| $c_l = 3$ | $62.27$ | $64.70$ | $67.13$ | $67.54$ | $70.20$ |
| $c_l = 4$ | $60.50$ | $60.62$ | $62.87$ | $65.10$ | $65.97$ |
| $c_l = 2$ | $73.35$ | $73.77$ | $74.69$ | $84.43$ | $87.23$ |
| $c_l = 3$ | $70.88$ | $72.38$ | $72.06$ | $73.75$ | $77.61$ |
| $c_l = 4$ | $69.30$ | $69.05$ | $70.39$ | $73.77$ | $73.86$ |

strategy by considering the relevancy of each label pair in the candidate label set.

8) **SURE** [36]: A self-training-based unified framework that utilizes the maximum infinity norm regularization to jointly train the predictive model and perform pseudo-labeling (ID type).

9) **PL-AGGD** [37]: Feature-aware disambiguation that jointly identifies the ground-truth label, determines similarity graph, and learns the predictive model (ID type).

To measure the generalization performance of the baselines and SGPPSL, we employ a fivefold cross-validation strategy to train these models on the selected datasets and report the average performance. Specifically, for SGPPSL cross-validation is set as an outer loop of the framework of parameter estimation as described in Section III-D.

### C. Recovering the Ground-Truth Label

The estimated confidence measure $C$ can be used to determine the ground-truth label for each token in the sequence, where $y_i = \arg \max_{c_j \in c_i} c_j$. Table II records the average accuracy of SGPPSL on recovering the ground-truth label for training data. With the increase of proportion of annotated instances, in most cases, the accuracy of recovering the ground-truth label is improved. Furthermore, the performance is negatively corrected with the number of candidate labels. Multiple annotations based on diverse backgrounds increase label ambiguity.

To demonstrate the confidence evolution in alternating optimization, we randomly select a group from fivefold cross-validation configurations. We record the change of confidence values of candidate labels for some training instances under several special settings: in base NP “holding” from the sentence “…is little holding sterling firm at…” with [{“B”}, “O”, “I”], $p = 0.1$. “German” from the sentence “…it disagreed with German advice…” with [{“B”–“LOC”}, “B-MISC”, “B–“ORG”}], $p = 0.5$ for NER, and in chunking “heavily” from the sentence “…figure are very heavily on…” with [{“I-ADVP”}, “B–“ADV P””, “B–“PRT”}], $p = 0.5$, where the ground-truth label is highlighted in bold.

As shown in Fig. 5, we can see that “holding” can be correctly assigned with “I” even with little exactly annotated training samples in base NP, while “German” and “heavily” are misclassified with half exact annotations. Compared with a relatively large label set, limited label space (e.g., base NP task) does not add too much noise in partial annotations. This helps reduce confusing label information for disambiguation. Furthermore, existing partial sequence labeling models employ a unique identification strategy, which may select the wrong ground-truth label for parameter learning. The proposed SGPPSL introduces confidence measure to address different contributions of candidate labels, which enables the ground-truth label to be utilized in the learning process.

### D. Comparing With the Baselines

By varying $c_l$ from 2 to 4 and $p$ from 0.1 to 0.9, we report the average F1 score that has been widely used in NLP tasks [15] to measure the prediction performance on test data.

1) **Base NP**: Fig. 6 presents the performance of compared methods on the base NP task. We can observe that SGPPSL outperforms the most of baselines. PL-SVM, PL-AGGD, and SURE achieve competitive results with lower $p$ (e.g., $p = 0.1$). While the average performance of $p > 0.5$ is always better than that of $p < 0.5$, there is no significant positive (or negative) relationship between the performance and the proportion of exactly annotated training instances for most of the PLL-based methods. Furthermore, the performance of PSL-based methods does not vary significantly as the number of candidate labels increases.

It is worth noting that the performance of most of the compared methods is very close. For example, SGPPSL outperforms the baselines in cl = 2 setting by a small margin. As the size of label set for base NP task is very small (i.e., [“B,” “I,” “O”]), there is not much confusing label information in partial annotations, which may decrease disambiguating difficulty for PSL models. Thus, most of the compared methods can achieve good performance even with lower $p$.

2) **Chunking**: Fig. 7 shows the performance of compared methods on the chunking task. It is obvious that SGPPSL consistently outperforms the other compared methods. By increasing the proportion of annotated training instances, SGPPSL, PL-SVM, and CLPL perform more stably than the other baselines. When $c_l = 4$, the performance of PSL-based models is positively correlated with the proportion of exactly annotated sequences. Moreover, with the increase of candidate labels, the performance with lower $p$ is decreased.

Different from the base NP task, the size of label set for chunking is relatively large (as described in Table 1). SGPPSL addresses different contributions of candidate labels in the learning process, which not only avoids wrong assignment...
of the ground truth but takes the confusing label information with inherent ambiguity of language into account. For example, “holding” in the sentence “…seven directors of each holding company…” has three candidate labels \{'I-NP', ‘I−VP’, ‘B−VP’\} (ground truth is highlighted in bold). As “holding” can be “B-VP” in the sentence “…there is little holding sterling firm…,” learning the specific weight to the pair \{“holding” → “B-VP”\} also expresses how confusing the label pair \{“I-NP,” “B-VP”\} to the word “holding,” which can help improve the performance of the model in identifying the similar word as “I-NP” or “B-VP.”

3) NER: Fig. 8 presents the results on NER task. It can be seen that SGPPSL outperforms the other compared methods when \(p < 0.7\). CLPL shows its superiority with the very close results compared with that of SGPPSL. When \(cl = 4\), these PSL-based methods can achieve better results by increasing the proportion of exactly annotated sequences.

There is also confusing label information in the NER task. For example, “European” is tagged as “B-MISC” in the sentence “…of their European Champions Cup…” while “B-ORG” in the sentence “…membership of the European Union…” Although the size of label set for NER is relatively small, confidence-weighted mechanism enables that SGPPSL achieves better performance with lower \(p\).

Based on the above observations on the performance of three tasks, SGPPSL is more effective in handling partial annotations with much confusing label information. While PL-SVM and CLPL obtain competitive results in some settings, average and unique disambiguation strategy may ignore the contribution of the ground-truth label and inherent ambiguity in the language, which may adversely affect the performance of NLP tasks. Generally, increasing the proportion of exactly annotated instance can help uncover the ground-truth labels and thus enhance the performance.

E. Analyzing Different Sampling Toward Generating Candidate Label Set

In this article, we generate a candidate label set with size \(cl\) by conducting random sampling \(cl\) times and each randomly select a negative label to the candidates. To enable different label partialities among partially labeled training samples, we employ the sampling strategy as described in [38] by controlling the flipping probability \(r\) that a negative label is flipped to the candidate label set. For this sampling strategy, we choose less-partial \(r = 0.05\) and strong-partial \(r = 0.75\) scenarios with \(p = 0.5\). Also, to give a more intuitive comparison, we report the results of \(cl = 2\) with \(p = 0.5\) presented in Section IV-D.

Fig. 9 shows the performance with different sampling toward generating candidate label set. We can see that in most cases, the results of \(r = 0.05\) are consistent with the results of \(cl = 2\), as in a less-partial setting, most of the partially annotated samples have only one additional negative label, which inevitably results in lower performance compared with the results of \(r = 0.05\). Furthermore, as the label space is limited in base NP (i.e., \(|\mathcal{Y}| = 3\)), the performance with \(r = 0.75\) approximates to the results of \(cl = 3\).
To verify the proposed weighted Viterbi algorithm for prediction, we compare it with the traditional Viterbi, where $cl = 3$ as the SGPPSL performs stably with varying $cl$. Table III reports the performance of different prediction algorithms. As shown in Table III, the performance of both decoding algorithms is positively correlated with the proportion of exactly annotated instances. Furthermore, in most cases, weighted Viterbi outperforms Viterbi, especially with lower $p$.

With the large proportion of ambiguous annotations, the uncertainty of label assignment in prediction is greatly increased. The weighted Viterbi algorithm incorporates confidence measures estimated in the training process to reduce the uncertainty arising from ambiguous annotations. For example, as shown in Fig. 3, supposing that the set of ambiguous annotations of the neighborhoods of the word “lead” is [“NN,” “NNP,” “VBD,” “VBP”] when assigning the POS tag to “lead” in the decoding, the ground-truth label “NN” obtains the highest score as, in most cases, our proposed model can identify the ground truth with the greatest confidence measure.

V. CONCLUSION

In this article, we propose a nonparametric Bayesian model SGPPSL for partial sequence labeling. The proposed SGPPSL employs factor-as-piece likelihood approximation and confidence measure for each candidate label, which effectively avoids handling a large number of candidate output sequences generated by partially annotated data and addresses the different contributions of each candidate label. A weighted Viterbi algorithm is proposed to incorporate confidence measure in prediction. We conduct the experiments on the tasks: base NP, chunking, and NER. The experimental results show that the proposed SGPPSL is more effective in handling partial annotations with much confusing label information. Furthermore, the weighted Viterbi achieves better performance than the traditional Viterbi. In the future, we will consider efficient variational inference for structured GPs.

F. Evaluating Prediction Algorithms

TABLE III

| Task   | Prediction algorithm | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 |
|--------|----------------------|-----|-----|-----|-----|-----|
| Base NP| Viterbi              | 69.53 | 71.80 | 76.18 | 86.62 | 82.95 |
|        | Weighted Viterbi     | 72.40 | 75.28 | 77.92 | 79.36 | 82.50 |
| Chunking| Viterbi              | 69.30 | 70.54 | 74.69 | 79.24 | 80.37 |
|        | Weighted Viterbi     | 75.40 | 76.22 | 78.29 | 78.85 | 81.20 |
| NER    | Viterbi              | 72.91 | 75.70 | 81.35 | 85.87 | 85.96 |
|        | Weighted Viterbi     | 80.42 | 80.61 | 83.59 | 84.20 | 85.29 |

**REFERENCES**

[1] A. PVS and G. Karthik, “Part-of-speech tagging and chunking using conditional random fields and transformation based learning,” in Shallow Parsing for South Asian Languages, vol. 21. Hyderabad, India: IJCAI, 2007.

[2] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., May 2013, pp. 6645–6649.

[3] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF models for sequence tagging,” 2015, arXiv:1508.01991.
[4] I. Tschantzari, T. Hofmann, T. Joachims, and Y. Altun, “Support vector machine learning for interdependent and structured output spaces,” in Proc. 21st Int. Conf. Mach. Learn. (ICML), 2004, p. 104.

[5] X. Chu et al., “CRF-CNN: Modeling structured information in human pose estimation,” in Proc. Adv. Neural Inf. Process. Syst., pp. 316–324, 2016.

[6] R. Alizadeh, C. Caragea, and C. L. Giles, "Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents," in Proc. World Wide Web Conf. (WWW), 2019, pp. 2551–2557.

[7] F. Jiao, S. Wang, C.-H. Lee, R. Greiner, and D. Schuurmans, “Semi-supervised conditional random fields for improved sequence segmentation and labeling,” in Proc. 21st Int. Conf. Comput. Linguistics 44th Annu. Meeting (ACL ACL), 2006, pp. 209–216.

[8] G. Li, J. Zhang, and Z. Chen, “Structured output learning with candidate labels for local parts,” in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases. Cham: Springer, 2013, pp. 336–352.

[9] X. Lou and F. A. Hamprecht, “Structured learning from partial annotations,” in Proc. 29th Int. Conf. Intell. Conf. Mach. Learn., 2012, pp. 371–378.

[10] M. Abdar et al., “A review of uncertainty quantification in deep learning: Techniques, applications and challenges,” 2020, arXiv:2011.06225.

[11] B. H. Juang and L. R. Rabiner, “Hidden Markov models for speech recognition,” Technometrics, vol. 33, no. 3, pp. 251–272, Aug. 1991.

[12] Y. Cao, Y. Li, S. Coleman, A. Belatreche, and T. M. McGinnity, “Adaptive hidden Markov model with anomaly states for price manipulation detection,” IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 2, pp. 318–330, Feb. 2015.

[13] J. Laflerty, A. McCallum, and F. C. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in Proc. Int. Conf. Mach. Learn., 2001, pp. 282–289.

[14] M. Kim, “Mixtures of conditional random fields for improved structured output prediction,” IEEE Trans. Neural Netw. Learn. Syst., vol. 28, no. 5, pp. 1233–1240, May 2017.

[15] S. Sarawagi and W. W. Cohen, “Semi-markov conditional random fields for information extraction,” in Proc. Adv. Neural Inf. Process. Syst., 2005, pp. 1185–1192.

[16] X. Chu et al., “CRF-CNN: Modeling structured information in human pose estimation,” in Proc. Adv. Neural Inf. Process. Syst., 2016, pp. 316–324.

[17] C. Dong, J. Zhang, C. Zong, M. Hatori, and H. Di, “Character-based LSTM-CRF with radical-level features for Chinese named entity recognition,” in Natural Language Understanding and Intelligent Applications. Cham, Switzerland: Springer, 2016, pp. 239–250.

[18] U. Brefeld and T. Scheffer, “Semi-supervised learning for structured output variables,” in Proc. 23rd Int. Conf. Mach. Learn. (ICML), 2006, pp. 145–152.

[19] T. Cao, B. Sapp, and B. Taskar, “Learning from partial labels,” J. Mach. Learn. Res., vol. 12, pp. 1501–1536, Apr. 2011.

[20] N. Nguyen and R. Caruana, “Classification with partial labels,” in Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD), 2008, pp. 551–559.

[21] F. Yu and M.-L. Zhang, “Maximum margin partial label learning,” in Proc. Asian Conf. Mach. Learn., 2016, pp. 96–111.

[22] R. Jin and Z. Ghahramani, “Learning with multiple labels,” in Proc. Adv. Neural Inf. Process. Syst., 2003, pp. 921–928.

[23] M.-L. Zhang, F. Yu, and C.-Z. Tang, “Disambiguation-free partial label learning,” IEEE Trans. Knowl. Data Eng., vol. 29, no. 10, pp. 2155–2167, Oct. 2017.

[24] E. Hüllermeier and J. Beringer, “Learning from ambiguously labeled data,” in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases. Cham: Springer, 2016, pp. 215–231.