

Data and text mining

PySeqLab: an open source Python package for sequence labeling and segmentation

Ahmed Allam^{1,*} and Michael Krauthammer^{1,2,*}

¹Department of Pathology, Yale School of Medicine, New Haven, CT 06511, USA and ²Program in Computational Biology and Bioinformatics, Yale University, New Haven, CT 06520, USA

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on January 13, 2017; revised on May 31, 2017; editorial decision on July 9, 2017; accepted on July 19, 2017

Abstract

Motivation: Text and genomic data are composed of sequential tokens, such as words and nucleotides that give rise to higher order syntactic constructs. In this work, we aim at providing a comprehensive Python library implementing conditional random fields (CRFs), a class of probabilistic graphical models, for robust prediction of these constructs from sequential data.

Results: Python Sequence Labeling (PySeqLab) is an open source package for performing supervised learning in structured prediction tasks. It implements CRFs models, that is discriminative models from (i) first-order to higher-order linear-chain CRFs, and from (ii) first-order to higher-order semi-Markov CRFs (semi-CRFs). Moreover, it provides multiple learning algorithms for estimating model parameters such as (i) stochastic gradient descent (SGD) and its multiple variations, (ii) structured perceptron with multiple averaging schemes supporting exact and inexact search using ‘violation-fixing’ framework, (iii) search-based probabilistic online learning algorithm (SAPO) and (iv) an interface for Broyden–Fletcher–Goldfarb–Shanno (BFGS) and the limited-memory BFGS algorithms. Viterbi and Viterbi A* are used for inference and decoding of sequences. Using PySeqLab, we built models (classifiers) and evaluated their performance in three different domains: (i) biomedical Natural language processing (NLP), (ii) predictive DNA sequence analysis and (iii) Human activity recognition (HAR). *State-of-the-art* performance comparable to machine-learning based systems was achieved in the three domains without feature engineering or the use of knowledge sources.

Availability and implementation: PySeqLab is available through https://bitbucket.org/A_2/pyseqlab with tutorials and documentation.

Contact: ahmed.allam@yale.edu or michael.krauthammer@yale.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Sequence labeling is a crucial task in domains such as natural language processing (NLP) and bioinformatics. Given a sequence of observations (words, nucleotides), the goal is to tag/label each observation using a set of permissible tags, which represent higher order syntactic constructs such as part-of-speech or exon boundaries. A related task is sequence segmentation, which consists of predicting constructs composed of several observations, such as exons that are composed of multiple nucleotides. The underlying structure

of both the input and the output (i.e. sequence of observations such as words and its corresponding part-of-speech) is exploited to build better predictors/classifiers in a supervised learning paradigm. Because of this inherent structure, many of the developed models and algorithms in the literature are described as *structured prediction* tasks. Early notable models in this area are conditional random fields (CRFs) (Lafferty *et al.*, 2001). CRFs are ‘undirected’ graphical models that are ‘discriminative’ (i.e. models the conditional probability of the entire label sequence given the observation sequence)

and ‘global’ (i.e. uses feature vector mapping that considers the whole observation sequence with its corresponding label sequence). These characteristics together with the use of a log-linear model gave CRF an advantageous position over earlier models such as the hidden Markov models (HMMs) and the maximum-entropy Markov models (MEMMs). Another class of models that represents a generalization to CRFs is the semi-Markov CRFs (semi-CRFs) (Sarawagi and Cohen, 2004). Semi-CRFs tackle sequence segmentation by predicting tags that extend across several consecutive observations of the input sequence. Hence, CRFs could be seen as a special case of semi-CRFs when the segment length is 1 (i.e. each label is assigned to one observation). Existing literature on both classes of models is focused on linear-chain versions using the first-order Markov assumption. This simplification guarantees the tractability of the model training (i.e. estimating the parameters using exact inference) by using the sum-product algorithm (i.e. performing a variation of the forward-backward algorithm). In its original formulation, the linear-chain first-order Markov assumption restricts the applicability of CRFs to learning on adjacent pairs of label features (i.e. models that depend on two states; the current and the previous state), where increasing the model order (i.e. $k \geq 2$) would lead to exponential computational complexity in terms of k .

However, recent work by (Cuong *et al.*, 2014), showed under the assumption of *label pattern sparsity* that the use of higher-order models (i.e. models with $k \geq 2$) is feasible without incurring an exponential complexity in the training and inference algorithms of both CRFs and Semi-CRFs. We refer to these generalized models by HO-semiCRFs (Cuong *et al.*, 2014).

Generally, these probabilistic models are trained (i.e. the process of finding optimal weights) by optimizing the objective function that consists of the sum of the log-likelihood of the sequences in the training set. Typically, gradient computation is a prerequisite for performing such probabilistic optimization. However, alternative approaches for discriminative training exist, including search- and perceptron-based methods that are adapted for structured prediction task such as the *structured perceptron* (Collins, 2002). To obtain the advantages of both approaches (probabilistic- and search- based), a hybrid method (search-based probabilistic online learning, SAPO) (Sun, 2015) was recently proposed.

The PySeqLab package features the implementation of CRFs and semi-CRFs models supporting higher order features, as well as multiple optimization/training and inference methods, achieving state-of-the-art performance on structured prediction tasks.

2 Models and implementation features

PySeqLab includes an implementation of (1) the original first-order CRF (FO-CRF) formulation (Lafferty *et al.*, 2001), (2) higher-order CRF (HO-CRF) (Cuong *et al.*, 2014; Ye *et al.*, 2009) and (3) HO-semiCRF (Cuong *et al.*, 2014). In addition, variants of both HO-CRF and HO-semiCRF models implementing an efficient algorithm for gradient computation (i.e. efficient backward algorithm) as proposed in (Vieira *et al.*, 2016) are also provided. Gradient-based training methods are implemented such as (i) stochastic gradient descent (We used stochastic gradient ascent, as the objective is to maximize the log-likelihood of the sequences in training data). (Bottou and Le Cun, 2004) supporting adaptive learning rates (such as ADADELTA (Zeiler, 2012)) and multiple learning rate scheduling, (ii) variance reduction method using stochastic variance reduced gradient (SVRG) (Johnson and Zhang, 2013) and (iii) an interface to BFGS and limited-memory BFGS (LBFGS) (BFGS and limited-memory BFGS

are offered using the `scipy.optimize` module in the SciPy package) optimization routines that use the computed gradients in addition to second order information (estimation of hessian matrix) to optimize weights during training. Perceptron-based training is offered through structured perceptron with the support of multiple averaging schemes (Collins, 2002). The package also implements the hybrid SAPO (An adapted version of SAPO where the regularization is based on weight averaging as in structured perceptron case) (Sun, 2015) optimization. Sequence decoding is achieved using Viterbi algorithm (Viterbi, 1967) and Viterbi A* (Soong and Huang, 1990) making it possible to output top-k sequences. Additionally, inexact search is supported using beam search (i.e. pruning states falling off a specified beam size) allowing for faster inference and training is supported within the ‘violation-fixing’ framework (see (Huang *et al.*, 2012) for more details). Maximum likelihood (MLE) and maximum a posteriori (MAP) estimation are implemented by offering two regularization schemes: (i) L2 regularization (i.e. assuming prior Gaussian distribution on the model weights) and (ii) L1 regularization using the approach in (Tsuruoka *et al.*, 2009). A training workflow in addition to various utilities that operate on the dataset (i.e. data splitting, preprocessing and normalizing) and observation/feature functions that automatically extract attributes and generates features using user-provided feature templates are also provided. Measuring trained models’ performance is also supported using precision, recall, accuracy and F-measure.

3 Results

To demonstrate the use and potential of the PySeqLab package in structured prediction tasks, we evaluated its performance in three different domains: (i) Natural language processing (NLP), classifying terms in molecular biology texts according to the Bio-Entity Recognition task (Bio-NER) (Kim *et al.*, 2004), (ii) DNA sequence analysis, predicting Eukaryotic splice-junctions based on a publicly available dataset (Noordewier *et al.*, 1991) and (iii) Human activity recognition (HAR), recognizing locomotion and gestures from sensor data using the OPPORTUNITY challenge dataset (Chavarriaga *et al.*, 2013). We discuss model features, training and evaluation in the Supplementary Materials. Overall, the trained models achieved *state-of-the-art* performance (see Supplementary Materials) compared to existing machine-learning based systems, notably without using feature engineering or external knowledge sources. We make the source code publicly available, and provide online full instructions to use our code and trained models in the three focus domains.

4 Conclusion

We presented PySeqLab, a comprehensive Python package aimed at building robust models for labeling sequences. We demonstrated the utility of the package in three different domains. More generally, given a training data composed of sequences of observations and associated labels, PySeqLab will learn state-of-the-art models that are accessible to use, customize and experiment with.

Funding

AA is supported by grant number P2TIP1_161635 awarded by the Swiss National Science Foundation. MK is supported by P01 CA016038 and P50 CA121974, both from the US National Cancer Institute.

Conflict of Interest: none declared.

References

- Bottou, L. and Le Cun, Y. (2004) Large scale online learning. *Adv. Neural Inf. Process. Syst.*, **16**, 217–225.
- Chavarriga, R. et al. (2013) The Opportunity challenge: a benchmark database for on-body sensor-based activity recognition. *Pattern Recogn. Lett.*, **34**, 2033–2042.
- Collins, M. (2002) Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing – EMNLP '02*, pp. 1–8.
- Cuong, V.N. et al. (2014) Conditional random field with high-order dependencies for sequence labeling and segmentation. *J. Mach. Learn. Res.*, **15**, 981–1009.
- Huang, L. et al. (2012) Structured perceptron with inexact search. In: *2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–151.
- Johnson, R. and Zhang, T. (2013) Accelerating stochastic gradient descent using predictive variance reduction. *Adv. Neural Inf. Process. Syst.*, **26**, 1, 315–323.
- Kim, J.-D. et al. (2004) Introduction to the Bio-entity Recognition Task at JNLPBA. In: *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, pp. 70–75.
- Lafferty, J. et al. (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: *ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning*, 8(June), pp. 282–289.
- Noordewier, M.O. et al. (1991) Training knowledge-based neural networks to recognize genes in DNA sequences. *Adv. Neural Inf. Process. Syst.*, **3**, 530–536.
- Sarawagi, S. and Cohen, W.W. (2004) Semi-Markov conditional random fields for information extraction. In: Saul, L.K. et al. (eds.), *Proceedings of the 17th International Conference on Neural Information Processing Systems (NIPS'04)*, MIT Press, Cambridge, MA, USA, 1185–1192.
- Soong, F.K. and Huang, E.-F. (1990) A tree-trellis based fast search for finding the N Best sentence hypotheses in continuous speech recognition. In: *Proceedings of the Workshop on Speech and Natural Language – HLT '90*, pp. 12–19.
- Sun, X. (2015) *Towards Shockingly Easy Structured Classification: A Search-based Probabilistic Online Learning Framework*. <https://arxiv.org/abs/1503.08381>.
- Tsuruoka, Y. et al. (2009) Stochastic gradient descent training for L1-regularized log-linear models with cumulative penalty. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, **1**, p. 477.
- Vieira, T. et al. (2016) Speed-Accuracy Tradeoffs in Tagging with Variable-Order CRFs and Structured Sparsity. In: *EMNLP*.
- Viterbi, A. (1967) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory*, **13**, 260–269.
- Ye, N. et al. (2009) Conditional Random Fields with High-Order Features for Sequence Labeling. *Neural Inf. Process. Syst.*, **2**, 2.
- Zeiler, M.D. (2012) *ADADELTA: An Adaptive Learning Rate Method*. <https://arxiv.org/abs/1212.5701>.