WIKIPEDIA

Conditional random field

Conditional random fields (**CRFs**) are a class of statistical modeling method often applied in pattern recognition and machine learning and used for structured prediction. CRFs fall into the sequence modeling family. Whereas a discrete classifier predicts a label for a single sample without considering "neighboring" samples, a CRF can take context into account; e.g., the linear chain CRF (which is popular in natural language processing) predicts sequences of labels for sequences of input samples.

CRFs are a type of discriminative undirected probabilistic graphical model. They are used to encode known relationships between observations and construct consistent interpretations and are often used for labeling or parsing of sequential data, such as natural language processing or biological sequences[1] and in computer vision.[2] Specifically, CRFs find applications in POS Tagging, shallow parsing,[3] named entity recognition,[4] gene finding and peptide critical functional region finding,[5] among other tasks, being an alternative to the related hidden Markov models (HMMs). In computer vision, CRFs are often used for object recognition[6] and image segmentation.

Contents

Description

Lafferty, McCallum and Pereira[1] define a CRF on observations $\boldsymbol{X}$ and random variables $\boldsymbol{Y}$ as follows:

> Let $G = (V, E)$ be a graph such that
>
> $\boldsymbol{Y} = (\boldsymbol{Y}_v)_{v \in V}$, so that $\boldsymbol{Y}$ is indexed by the vertices of G. Then $(\boldsymbol{X}, \boldsymbol{Y})$ is a conditional random field when the random variables $\boldsymbol{Y}_v$, conditioned on $\boldsymbol{X}$, obey the Markov property with respect to the graph: $p(\boldsymbol{Y}_v | \boldsymbol{X}, \boldsymbol{Y}_w, w \neq v) = p(\boldsymbol{Y}_v | \boldsymbol{X}, \boldsymbol{Y}_w, w \sim v)$, where $w \sim v$ means that w and v are neighbors in G.

What this means is that a CRF is an undirected graphical model whose nodes can be divided into exactly two disjoint sets $\boldsymbol{X}$ and $\boldsymbol{Y}$, the observed and output variables, respectively; the conditional distribution $p(\boldsymbol{Y}|\boldsymbol{X})$ is then modeled.

Inference

For general graphs, the problem of exact inference in CRFs is intractable. The inference problem for a CRF is basically the same as for an MRF and the same arguments hold.[7] However, there exist special cases for which exact inference is feasible:

- If the graph is a chain or a tree, message passing algorithms yield exact solutions. The algorithms used in these cases are analogous to the forward-backward and Viterbi algorithm for the case of HMMs.
- If the CRF only contains pair-wise potentials and the energy is submodular, combinatorial min cut/max flow algorithms yield exact solutions.

If exact inference is impossible, several algorithms can be used to obtain approximate solutions. These include:

- Loopy belief propagation
- Alpha expansion
- Mean field inference
- Linear programming relaxations

Parameter Learning

Learning the parameters θ is usually done by maximum likelihood learning for $p(Y_i|X_i;\theta)$. If all nodes have exponential family distributions and all nodes are observed during training, this optimization is convex.[7] It can be solved for example using gradient descent algorithms, or Quasi-Newton methods such as the L-BFGS algorithm. On the other hand, if some variables are unobserved, the inference problem has to be solved for these variables. Exact inference is intractable in general graphs, so approximations have to be used.

Examples

In sequence modeling, the graph of interest is usually a chain graph. An input sequence of observed variables X represents a sequence of observations and Y represents a hidden (or unknown) state variable that needs to be inferred given the observations. The Y_i are structured to form a chain, with an edge between each Y_{i-1} and Y_i. As well as having a simple interpretation of the Y_i as "labels" for each element in the input sequence, this layout admits efficient algorithms for:

- model *training*, learning the conditional distributions between the Y_i and feature functions from some corpus of training data.
- *decoding*, determining the probability of a given label sequence Y given X.
- *inference*, determining the *most likely* label sequence Y given X.

The conditional dependency of each Y_i on X is defined through a fixed set of *feature functions* of the form $f(i, Y_{i-1}, Y_i, X)$, which can be thought of as measurements on the input sequence that partially determine the likelihood of each possible value for Y_i. The model assigns each feature a numerical weight and combines them to determine the probability of a certain value for Y_i.

Linear-chain CRFs have many of the same applications as conceptually simpler hidden Markov models (HMMs), but relax certain assumptions about the input and output sequence distributions. An HMM can loosely be understood as a CRF with very specific feature functions that use constant probabilities to model state transitions and emissions. Conversely, a CRF can loosely be understood as a generalization of an HMM that makes the constant transition probabilities into arbitrary functions that vary across the positions in the sequence of hidden states, depending on the input sequence.

Notably, in contrast to HMMs, CRFs can contain any number of feature functions, the feature functions can inspect the entire input sequence X at any point during inference, and the range of the feature functions need not have a probabilistic interpretation.

Variants

Higher-order CRFs and semi-Markov CRFs

CRFs can be extended into higher order models by making each Y_i dependent on a fixed number k of previous variables $Y_{i-k}, \ldots, Y_{i-1}$. In conventional formulations of higher order CRFs, training and inference are only practical for small values of k (such as $k \le 5$),[8] since their computational cost increases exponentially with k.

However, another recent advance has managed to ameliorate these issues by leveraging concepts and tools from the field of Bayesian nonparametrics. Specifically, the CRF-infinity approach[9] constitutes a CRF-type model that is capable of learning infinitely-long temporal dynamics in a scalable fashion. This is effected by introducing a novel potential function for CRFs that is based on the Sequence Memoizer (SM), a nonparametric Bayesian model for learning infinitely-long dynamics in sequential observations.[10] To render such a model computationally tractable, CRF-infinity employs a mean-field approximation [11] of the postulated novel potential functions (which are driven by an SM). This allows for devising efficient approximate training and inference algorithms for the model, without undermining its capability to capture and model temporal dependencies of arbitrary length.

There exists another generalization of CRFs, the **semi-Markov conditional random field (semi-CRF)**, which models variable-length *segmentations* of the label sequence Y.[12] This provides much of the power of higher-order CRFs to model long-range dependencies of the Y_i, at a reasonable computational cost.

Finally, large-margin models for structured prediction, such as the structured Support Vector Machine can be seen as an alternative training procedure to CRFs.

Latent-dynamic conditional random field

Latent-dynamic conditional random fields (**LDCRF**) or **discriminative probabilistic latent variable models** (**DPLVM**) are a type of CRFs for sequence tagging tasks. They are latent variable models that are trained discriminatively.

In an LDCRF, like in any sequence tagging task, given a sequence of observations $\mathbf{x} = x_1, \ldots, x_n$, the main problem the model must solve is how to assign a sequence of labels $\mathbf{y} = y_1, \ldots, y_n$ from one finite set of labels Y. Instead of directly modeling $P(\mathbf{y}|\mathbf{x})$ as an ordinary linear-chain CRF would do, a set of latent variables $\mathbf{h}$ is "inserted" between $\mathbf{x}$ and $\mathbf{y}$ using the chain rule of probability:[13]

$$P(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{h}} P(\mathbf{y}|\mathbf{h},\mathbf{x})P(\mathbf{h}|\mathbf{x})$$

This allows capturing latent structure between the observations and labels.[14] While LDCRFs can be trained using quasi-Newton methods, a specialized version of the perceptron algorithm called the **latent-variable perceptron** has been developed for them as well, based on Collins' structured perceptron algorithm.[13] These models find applications in computer vision, specifically gesture recognition from video streams[14] and shallow parsing.[13]

Software

This is a partial list of software that implement generic CRF tools.

- RNNSharp (https://github.com/zhongkaifu/RNNSharp) CRFs based on recurrent neural networks (C#, .NET)
- CRF-ADF (http://klcl.pku.edu.cn/member/sunxu/code.htm) Linear-chain CRFs with fast online ADF training (C#, .NET)
- CRFSharp (https://github.com/zhongkaifu/CRFSharp) Linear-chain CRFs (C#, .NET)
- GCO (http://vision.csd.uwo.ca/code/) CRFs with submodular energy functions (C++, Matlab)
- DGM (http://research.project-10.de/dgm) General CRFs (C++)
- GRMM (http://mallet.cs.umass.edu/grmm/index.php) General CRFs (Java)
- factorie (http://factorie.cs.umass.edu/) General CRFs (Scala)
- CRFall (http://www.cs.ubc.ca/~murphyk/Software/CRFall.zip) General CRFs (Matlab)
- Sarawagi's CRF (http://crf.sourceforge.net/) Linear-chain CRFs (Java)
- HCRF library (http://sourceforge.net/projects/hcrf/) Hidden-state CRFs (C++, Matlab)
- Accord.NET (http://accord-framework.net) Linear-chain CRF, HCRF and HMMs (C#, .NET)
- Wapiti (http://wapiti.limsi.fr/) Fast linear-chain CRFs (C)[15]
- CRFSuite (http://www.chokkan.org/software/crfsuite/) Fast restricted linear-chain CRFs (C)
- CRF++ (https://web.archive.org/web/20100421020327/http://crfpp.sourceforge.net/) Linear-chain CRFs (C++)
- FlexCRFs (http://flexcrfs.sourceforge.net/) First-order and second-order Markov CRFs (C++)
- crf-chain1 (http://hackage.haskell.org/package/crf-chain1) First-order, linear-chain CRFs (Haskell)
- imageCRF (http://www.cs.rochester.edu/~bhole/code/crf/) CRF for segmenting images and image volumes (C++)
- MALLET (http://mallet.cs.umass.edu/) Linear-chain for sequence tagging (Java)
- PyStruct (https://pystruct.github.io/) Structured Learning in Python (Python)
- Pycrfsuite (https://github.com/scrapinghub/python-crfsuite) A python binding for crfsuite (Python)
- Figaro (https://github.com/p2t2/figaro) Probabilistic programming language capable of defining CRFs and other graphical models (Scala)
- CRF (https://cran.r-project.org/web/packages/CRF/CRF.pdf) Modeling and computational tools for CRFs and other undirected graphical models (R)
- OpenGM (http://hciweb2.iwr.uni-heidelberg.de/opengm/index.php) Library for discrete factor graph models and distributive operations on these models (C++)
- UPGMpp (https://github.com/jotaraul/upgmpp)[6] Library for building, training, and performing inference with Undirected Graphical Models (C++)
- KEG_CRF (http://keg.cs.tsinghua.edu.cn/jietang/software/KEG_CRF/) Fast Linear CRFs (C++)

This is a partial list of software that implement CRF related tools.

- MedaCy (https://github.com/NLPatVCU/medaCy) Medical Named Entity Recognizer (Python)
- Conrad (http://www.broadinstitute.org/annotation/conrad) CRF based gene predictor (Java)
- Stanford NER (http://nlp.stanford.edu/software/CRF-NER.shtml) Named Entity Recognizer (Java)
- BANNER (https://web.archive.org/web/20100707042144/http://cbioc.eas.asu.edu/banner/) Named Entity Recognizer (Java)

See also

- Hammersley–Clifford theorem
- Graphical model
- Markov random field
- Maximum entropy Markov model (MEMM)

References

1. Lafferty, J., McCallum, A., Pereira, F. (2001). "Conditional random fields: Probabilistic models for segmenting and

labeling sequence data" (http://repository.upenn.edu/cgi/viewcontent.cgi?article=1162&context=cis_papers). *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann. pp. 282–289.

2. He, X.; Zemel, R.S.; Carreira-Perpiñán, M.A. (2004). "Multiscale conditional random fields for image labeling". IEEE Computer Society. CiteSeerX 10.1.1.3.7826 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.3.7826).
3. Sha, F.; Pereira, F. (2003). *shallow parsing with conditional random fields* (http://portal.acm.org/ft_gateway.cfm?id=1073473&type=pdf&CFID=4684435&CFTOKEN=39459323).
4. Settles, B. (2004). "Biomedical named entity recognition using conditional random fields and rich feature sets" (http://www.aclweb.org/anthology/W04-1221.pdf) (PDF). *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*. pp. 104–107.
5. Chang KY; Lin T-p; Shih L-Y; Wang C-K (2015). *Analysis and Prediction of the Critical Regions of Antimicrobial Peptides Based on Conditional Random Fields*. PLoS ONE. doi:10.1371/journal.pone.0119490 (https://doi.org/10.1371%2Fjournal.pone.0119490). PMC 4372350 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4372350).
6. J.R. Ruiz-Sarmiento; C. Galindo; J. Gonzalez-Jimenez (2015). "UPGMpp: a Software Library for Contextual Object Recognition." (https://www.researchgate.net/publication/281620302_UPGMpp_a_Software_Library_for_Contextual_Object_Recognition). *3rd. Workshop on Recognition and Action for Scene Understanding (REACTS)*.
7. Sutton, Charles; McCallum, Andrew (2010). "An Introduction to Conditional Random Fields". arXiv:1011.4088v1 (https://arxiv.org/abs/1011.4088v1) [stat.ML (https://arxiv.org/archive/stat.ML)].
8. Lavergne, Thomas; Yvon, François (September 7, 2017). "Learning the Structure of Variable-Order CRFs: a Finite-State Perspective" (http://aclweb.org/anthology/D17-1044). *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics. p. 433.
9. Chatzis, Sotirios; Demiris, Yiannis (2013). "The Infinite-Order Conditional Random Field Model for Sequential Data Modeling". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **35** (6): 1523–1534. doi:10.1109/tpami.2012.208 (https://doi.org/10.1109%2Ftpami.2012.208). PMID 23599063 (https://www.ncbi.nlm.nih.gov/pubmed/23599063).
10. Gasthaus, Jan; Teh, Yee Whye (2010). "Improvements to the Sequence Memoizer" (https://papers.nips.cc/paper/3938-improvements-to-the-sequence-memoizer.pdf) (PDF). *Proc. NIPS*.
11. Celeux, G.; Forbes, F.; Peyrard, N. (2003). "EM Procedures Using Mean Field-Like Approximations for Markov Model-Based Image Segmentation". *Pattern Recognition*. **36** (1): 131–144. CiteSeerX 10.1.1.6.9064 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.6.9064). doi:10.1016/s0031-3203(02)00027-4 (https://doi.org/10.1016%2Fs0031-3203%2802%2900027-4).
12. Sarawagi, Sunita; Cohen, William W. (2005). "Semi-Markov conditional random fields for information extraction" (http://papers.nips.cc/paper/2648-semi-markov-conditional-random-fields-for-information-extraction.pdf) (PDF). In Lawrence K. Saul, Yair Weiss, Léon Bottou (eds.) (eds.). *Advances in Neural Information Processing Systems 17* (http://papers.nips.cc/book/advances-in-neural-information-processing-systems-17-2004). Cambridge, MA: MIT Press. pp. 1185–1192.
13. Xu Sun; Takuya Matsuzaki; Daisuke Okanohara; Jun'ichi Tsujii (2009). *Latent Variable Perceptron Algorithm for Structured Classification* (http://www.aaai.org/ocs/index.php/IJCAI/IJCAI-09/paper/download/356/970). IJCAI. pp. 1236–1242.
14. Morency, L. P.; Quattoni, A.; Darrell, T. (2007). "Latent-Dynamic Discriminative Models for Continuous Gesture Recognition" (http://dspace.mit.edu/bitstream/handle/1721.1/35276/MIT-CSAIL-TR-2007-002.pdf) (PDF). *2007 IEEE Conference on Computer Vision and Pattern Recognition*. p. 1. CiteSeerX 10.1.1.420.6836 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.420.6836). doi:10.1109/CVPR.2007.383299 (https://doi.org/10.1109%2FCVPR.2007.383299). ISBN 978-1-4244-1179-5.
15. T. Lavergne, O. Cappé and F. Yvon (2010). Practical very large scale CRFs (http://acl.eldoc.ub.rug.nl/mirror/P/P10/P10-1052.pdf) Archived (https://web.archive.org/web/20130718001211/http://acl.eldoc.ub.rug.nl/mirror/P/P10/P10-1052.pdf) 2013-07-18 at the Wayback Machine. Proc. 48th Annual Meeting of the ACL, pp. 504-513.

Further reading

- McCallum, A.: Efficiently inducing features of conditional random fields (https://arxiv.org/pdf/1212.2504). In: *Proc. 19th Conference on Uncertainty in Artificial Intelligence*. (2003)
- Wallach, H.M.: Conditional random fields: An introduction (http://www.cs.umass.edu/~wallach/technical_reports/wallach04conditional.pdf). Technical report MS-CIS-04-21, University of Pennsylvania (2004)
- Sutton, C., McCallum, A.: An Introduction to Conditional Random Fields for Relational Learning. In "Introduction to Statistical Relational Learning". Edited by Lise Getoor and Ben Taskar. MIT Press. (2006) Online PDF (http://www.cs.umass.edu/~mccallum/papers/crf-tutorial.pdf)
- Klinger, R., Tomanek, K.: Classical Probabilistic Models and Conditional Random Fields. Algorithm Engineering Report TR07-2-013, Department of Computer Science, Dortmund University of Technology, December 2007. ISSN 1864-4503. Online PDF (https://ls11-www.cs.uni-dortmund.de/_media/techreports/tr07-13.pdf)