Integer Linear Programming Inference for Conditional Random Fields

By Dan Roth and Wen-tau Yih, ICLM'05

Ran Chen

University of Pennsylvania

ran1chen@wharton.upenn.edu

October 18, 2017

Overview

- Incorporating general constraints over the output space is natural and important in many settings
- Though Viterbi, a dynamic programming algorithm can be used efficiently to output the labels that maximize the joint conditional probability given the observation (in CRF setting), it fails to encode general constraints.
- Integer linear programming can be used to incorporate a wide range of general constraints
- Experimentally, the post-training inference incorporating general constraints by integer linear programming dramatically improves the performance of both CRF based methods and local learning algorithms.

Overview

General Linear Chain CRF (Model)

- Assume there are K feature functions, $f^1, \cdots, f^K$, each of them maps a pair of sequence $(\mathbf{y}, \mathbf{x})$ and token index i to $f^k(\mathbf{y}, \mathbf{x}, i) \in \mathbb{R}$. Where $\mathbf{y}$ stands for the sequence of label and $\mathbf{x}$ stands for the sequence of observation.
- The global feature vector is defined by $F(\mathbf{y}, \mathbf{x}) = \sum_i < f^1(\mathbf{y}, \mathbf{x}, i), \cdots, f^K(\mathbf{y}, \mathbf{x}, i) >$
- the probability distribution is defined as

$$Pr_\lambda(\mathbf{Y}|\mathbf{X}) = \frac{exp(\lambda \cdot F(\mathbf{Y}, \mathbf{X}))}{Z_\lambda(\mathbf{X})}$$

  , where $Z_\lambda(\mathbf{X}) = \sum_{\mathbf{Y}} exp(\lambda \cdot F(\mathbf{Y}, \mathbf{X}))$ is a normalization factor
- The goal is to find $\mathbf{y}$ maximizing the above quantity.

Linear Chain CRF and Viterbi Algorithm (Inference) - 1

- When $f^k(\mathbf{y}, \mathbf{x}, i)$ is only related to $\mathbf{x}$, y_{i-1} and y_i , define $M_i(y', y|x) = exp(\sum_j \lambda_j f_j(y', y, \mathbf{x}, i))$, where $y', y \in \mathcal{Y}$
- The sequence probability is $p(\mathbf{y}|\mathbf{x}, \lambda) = \frac{1}{Z_\lambda(\mathbf{x})} \Pi_{i=0}^{n} M_i(y_{i-1}, y_i|\mathbf{x})$, where y_{-1}, y_n are two augmented special nodes before and after the **start** and **end** of the sequence.
- $\hat{\mathbf{y}} = \mathrm{argmax}_{\mathbf{y}} \frac{1}{Z_\lambda(\mathbf{x})} \Pi_{i=0}^{n} M_i(y_{i-1}, y_i|\mathbf{x}) = \mathrm{argmax}_{\mathbf{y}} \Pi_{i=0}^{n} M_i(y_{i-1}, y_i|\mathbf{x})$

Linear Chain CRF and Viterbi Algorithm (Inference) - 2

- **Viterbi Algorithm** computes the most likely label sequence ($\hat{\mathbf{y}}$) given the observation $\mathbf{x}$. At step i, it records all the optimal sequences ending at label y, $\forall y \in \mathcal{Y}$, $\mathbf{y}_i^*(y)$, and also the corresponding product $P_i(y)$.
- **The recursive function of Viterbi Algorithm**
  1. $P_0(y) = M_0(start, y|\mathbf{x})$ and $\mathbf{y}_0^*(y) = y$
  2. For $1 \leq i \leq n$, $\mathbf{y}_i^*(y) = \mathbf{y}_{i-1}^*(\hat{y}) \cdot (y)$ and $P_i(y) = max_{y' \in \mathcal{Y}} P_{i-1}(y')M(y', y|(x))$, where $\hat{y} = argmax_{y' \in \mathcal{Y}} P_{i-1} y' M(y', y|\mathbf{x})$ and "." is the concatenation operator

Training of Linear Chain CRFs

- Training is to estimate the values of the weight vector λ given the training set $T = \{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^{N}$, where $\mathbf{x}_k$ and $\mathbf{y}_k$ are observation sequence and label sequence.

$$\begin{aligned}\hat{\lambda} &= \operatorname*{argmax}_{\lambda} \mathcal{L}_{\lambda} = \operatorname*{argmax}_{\lambda} \sum_{k} log(p_{\lambda}(\mathbf{y}_k|\mathbf{x}_k)) \\ &= \operatorname*{argmax}_{\lambda} \sum_{k} [\lambda \cdot F(\mathbf{y}_k, \mathbf{x}_k) - \log Z_{\lambda}(\mathbf{x}_k)]\end{aligned}$$

- Training methods:
  1. **maximum likelihood training** Find $\hat{\lambda}$, e.g. generalized iterative scaling, conjugate-gradient, limited-memory quasi-Newton
  2. **discriminatively learning** reducing the number of error predictions directly (find $\tilde{\lambda} = \mathrm{argmax}_{\lambda} -|\{k : \mathbf{y_k} \neq \mathrm{argmax}_{\mathbf{y}}\, log(p_{\lambda}(\mathbf{y}|\mathbf{x_k}))\}|$), e.g. sturctured (Voted) perceptron (Collins 2002)

Incorporating Constraints in Viterbi—Natural Constraints on output spaces 1

- In many NLP problems (e.g. chunking, semantic role labeling, information extraction), there is a need to identify segments of consecutive words in the sentence and classify them to one of several classes
- **BIO representation**:
  - label B- (Begin): the first word of a segment, "-" indicates the phrase type
  - label I- (Inside): the word is part of , but not first in the segment
  - label O (Outside): all other words in the sentence

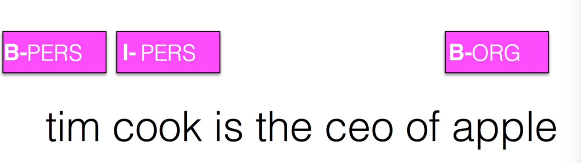


Figure: suppose we have types: person, location, time, money, organization

Incorporating Constraints in Viterbi—Natural Constraints on output spaces 2

- When no consecutive segments share the same type, BIO representation can be simplified to the **IO representation**
- **Constraints**: "no duplicate segments" (e.g. semantic role labeling); "I" does not follow "O" in BIO representation; a known label of a specific position or disallow some tokens

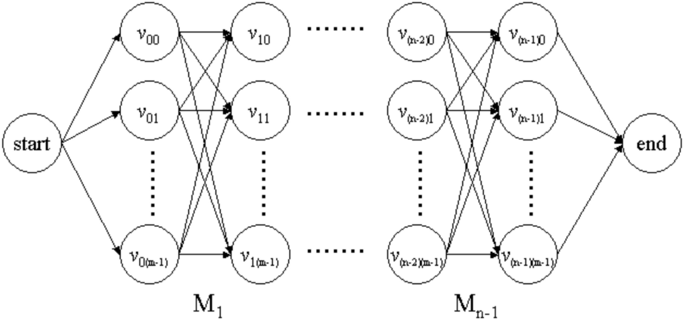
Incorporating Constrains in Viterbi

- I label does not immediately follow O label $\Rightarrow$ set $M_i(y_{i-1} = O, y_i = I|\mathbf{x}) = 0, \forall 1 \leq i \leq n-1$
- Label at position i has to be a $\Rightarrow$ set $M_i(y_{i-1}, y_i) = 0, \forall y_{i-1} \in \mathcal{Y}$ and all $y_i \in \mathcal{Y} - \{a\}$
- Global constraints? (e.g. no duplicate segments, relation between distant tokens) $\Rightarrow$ unable to incorporate in Viterbi

Modification of matrix M is not sufficient to incorporate long distant, more general constraints

Reformulatin gthi problem in Shortest Path Problem

- Graph G=(V,E)
- $mn + 2$ nodes: start, end, $v_{i,j}$ ($v_{i,j}$ i th point with j th label)
- $2m + (n-1)m^2$ edges: $(start, v_{0,j})$, $(v_{n-1,j}, end)$, $(v_{i,j}, v_{i+1,k})$
- edge weight: edge weight of $v_{i-1,y}$ and $v_{i,y'}$ is $-log(M_i(y, y'|\mathbf{x}))$
- reformulate to shortest path problem: find $argmax_{\mathbf{y}} \Pi_{i=0}^{n-1} M_i(y_{i-1}y_i|\mathbf{x})$ $\Leftrightarrow$ find the shortest path



Reformulating the problem in ILP Setting-1

The shortest path problem could be reformulated in integer linear programming

$$\min \sum_{(u,v)\in E} c_{uv} \cdot x_{uv}$$

subject to:

$$\begin{aligned}
\sum_{u\in V^-(v)} x_{uv} - \sum_{w\in V^+(v)} x_{vw} &= 0, \qquad \forall v \in V - \{s,t\} \\
\sum_{u\in V^-(s)} x_{us} - \sum_{w\in V^+(s)} x_{sw} &= -1 \\
\sum_{u\in V^-(t)} x_{ut} - \sum_{w\in V^+(t)} x_{tw} &= 1 \\
x_{uv} \in \{0,1\}, &\qquad \forall (u,v) \in E
\end{aligned}$$

Reformulating the problem in ILP Setting-2

Taking the objective function into the formulation, we have

$$\max \sum_{\substack{0 \le i \le n-1 \\ 0 \le y, y' \le m-1}} \log M_i(y, y') \cdot x_{i,yy'}$$

subject to:

$$\sum_{0 \le y_1 \le m-1} x_{i-1,y_1 y} - \sum_{0 \le y_2 \le m-1} x_{i,yy_2} = 0,$$

for all i, y such that $0 \le i \le n-1, 0 \le y \le m-1$.

$$\sum_{0 \le y \le m-1} x_{-1,0y} = 1 \text{ and } \sum_{0 \le y \le m-1} x_{n,y0} = 1$$

$$x_{-1,0y}, x_{i,y_1 y}, x_{n,y0} \in \{0, 1\},$$

for all i, y_1, y such that $0 \le i \le n-1, 0 \le y_1, y \le m-1$.

Incorporating global constraints through ILP-examples

- **no duplicate argument labels**

$$m(n-1-i)x_{i,ab} \leq \sum_{\substack{0\leq y\leq m-1 \\ i+1\leq j\leq n-1}} 1 - x_{j,ya}$$

, $\forall i, a, b$ s.t. $1 \leq i \leq n-2,\ 0 \leq b \leq m-1,\ a \neq b$

- **at least one Argument** At least one segment should not be O:

$$\sum_{\substack{0\leq i\leq n-1 \\ 0\leq y\leq m-1}} x_{i,y0} \leq n-1$$

- **Known verb position** The verb (at position i) should be labeled O:

$$\sum_{0\leq y\leq m-1} x_{i,y0} = 1$$

Incorporating global constraints through ILP-examples

- **segment $\mathcal{A}$ of tokens share the same label**
  Denote $v_{i,y} = \sum_{0 \leq y' \leq m-1} x_{i,y'y}$, then the constraint is

$$|\mathcal{A}| v_{p,l} \leq \sum_{i \in \mathcal{A}} v_{i,l}$$

  $\forall p \in \mathcal{A}, \forall l, 0 \leq l \leq m-1$

- **a appears $\Rightarrow$ b appears**

$$\sum_{0 \leq y \leq m-1} x_{j,ya} \leq \sum_{\substack{0 \leq y \leq m-1 \\ 0 \leq i \leq n-1}} x_{i,yb}$$

  $\forall j, 0 \leq j \leq n-1$

Incorporating global constraints in Integer Linear Prgramming Setting-2

Theorem

All possible Boolean functions over the variables of interest can be represented as sets of linear (in)equalities (Gueret et al., 2002)

Properties of Linear programming

Definition (TU)

A matrix A is *totally unimodular if the determinant of every square submatrix of* **A** *is* $+1$*,-1,0.*

Theorem (Veinott & Dantzig)

Let **A** *be an (m,n)-integral (integer) matrix with full row rank m. Then solution to the linear program max*$(\mathbf{cx} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \in \mathbb{R}^n_+)$ *is integral (integer) vector* **b***, if and only if* **A** *is totally unimodular.*

Linear programming results on Shortest Path Problem

Theorem (Wolsey, 1998)

The coefficient matrix of the linear programming for the shortest path problem is totally unimodular

$\Rightarrow$(Computing Complexity)

Shortest path problem could be transformed from integer linear programming into a common linear programming (adding constraints $x_{i,uv} \leq 1, -x_{i,uv} \leq 0$). Therefore, interior point could be used to solve it, which only takes polynomial time.

Experiment Setting

- **Semantic Role Labeling** using the definition of PropBank, taking only the core arguments. Using IO representation. The goal is to assign each chunk with one of the following labels: O, I-A0, I-A1, I-A2,I-A3,I-A4,I-A5

I	left	my	pearls	to	my	daughter-in-law	in	my	will	.
I-A0	O	I-A1	I-A1	I-A2	I-A2	I-A2	O	O	O	O
A0	V	A1		A2						

- features: state features; word; pos,chunk type of the neighboring chunk; edge; end
- general constraints are not used in training procedures, they are only use in inference procedure
- CRFs are trained through maximum log likelihood and discriminative method

Experiment Setting-Constraints

- **No duplicate argument labels**
  In the SRL task, a verb in a sentence cannot have two arguments of the same type.
- **Argument candidates**
  - Generate a candidate list with high recall but low precision.
  - Each candidate argument is a segment of consecutive chunks.
  - Although not every candidate is an argument of the target verb, each chunk in the candidate has to be assigned the same label.
  - This is an effective constraint that provides argument-level information.
- **At least one argument**
  Each verb in a sentence must have at least one core argument $\Rightarrow$ at least one chunk will be assigned a label other than O.
- **Known verb position**
  the known verbs should be labeled 0.
- **Disallow arguments**
  - Given a particular verb, not every argument type is legitimate.
  - The arguments that a verb can take are defined in the frame files in the PropBank corpus.

Experiment Results - CRFs

- General constraints improve the results significantly, for both CRF training algorithm.
- However, incorporating general constraints to discriminative training method (CRF-D) does not perform satisfactory, which may due to the limited training examples (Punykanok et al., 2005)

		CRF-ML			CRF-D		
		Rec	Prec	F_1	Rec	Prec	F_1
	basic	62.53	70.91	66.46	66.64	71.83	69.14
1	+ no dup	62.52	72.41	67.10	66.21	73.66	69.74
2	+ candidate	65.61	79.23	71.78	68.64	79.44	73.64
3	+ argument	66.54	77.76	71.71	69.42	78.57	73.71
4	+ verb pos	66.56	77.75	71.72	69.52	78.59	73.78
5	+ disallow	66.70	78.08	**71.94**	69.62	78.76	**73.91**

Experiment Results - Local Learning Systems

- To totally decoupling learning and inference, the training procedure could be reduced to multi-class classifier. Constraints are only used in the inference procedure.
- The multi-class classifiers used are: regularized versions of perceptron, winnow, voted perceptron and voted winnow. The criteria is F_1.
- The results shows that with the increasing constraints, local learning systems' performance improve dramatically and even surpass the CRFs when all 5 constraints are encompassed.

		VP	VW	P	W
	basic	58.15	54.32	53.03	50.78
1	+ no dup	64.33	61.87	60.59	59.13
2	+ candidate	74.17	71.72	70.03	70.20
3	+ argument	74.02	71.76	69.98	70.32
4	+ verb pos	74.03	71.84	70.05	70.42
5	+ disallow	**74.49**	72.04	70.36	70.67

Experiment Results - Comparison

- CRF results

		CRF-ML			CRF-D		
		Rec	Prec	F_1	Rec	Prec	F_1
	basic	62.53	70.91	66.46	66.64	71.83	69.14
1	+ no dup	62.52	72.41	67.10	66.21	73.66	69.74
2	+ candidate	65.61	79.23	71.78	68.64	79.44	73.64
3	+ argument	66.54	77.76	71.71	69.42	78.57	73.71
4	+ verb pos	66.56	77.75	71.72	69.52	78.59	73.78
5	+ disallow	66.70	78.08	**71.94**	69.62	78.76	**73.91**

- Local learning system's result

		VP	VW	P	W
	basic	58.15	54.32	53.03	50.78
1	+ no dup	64.33	61.87	60.59	59.13
2	+ candidate	74.17	71.72	70.03	70.20
3	+ argument	74.02	71.76	69.98	70.32
4	+ verb pos	74.03	71.84	70.05	70.42
5	+ disallow	**74.49**	72.04	70.36	70.67

- IBT, L+I, the difficulty of local training

Experiment Results - Run Time

Local learning based systems are more efficient (w.r.t runtime).

	CRF-ML	CRF-D	CRF-D (IBT)	VP	VW
Time (hrs)	48	38	145	0.8	0.8

Take-Away Points

- The shortest path problem solved by Viterbi algorithm can be represented and solved through integer linear programming
- Integer linear programming can systematically incorporate general constraints that can not be incorporated in Viterbi
- Experimentally, Incorporating general constraints through integer linear programming indeed dramatically improve the performances of both CRFs and local learning systems.
- Sometimes, in the presence of structure on output, enforcing the constrains only at the evaluation time results in comparable performance at a much lower cost.

References

Roth D, Yih W (2005)
Integer linear programming inference for conditional random fields[C]
Proceedings of the 22nd international conference on Machine learning ACM, 2005: 736-743.

Punyakanok, V., Roth, D., Yih, W., & Zimak, D. (2005)
Learning and inference over constrained output.
Proc. of IJCAI-2005.

Collins, M. (2002)
Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms.
Proc. of EMNLP-2002.

The End

Thanks for your attention!