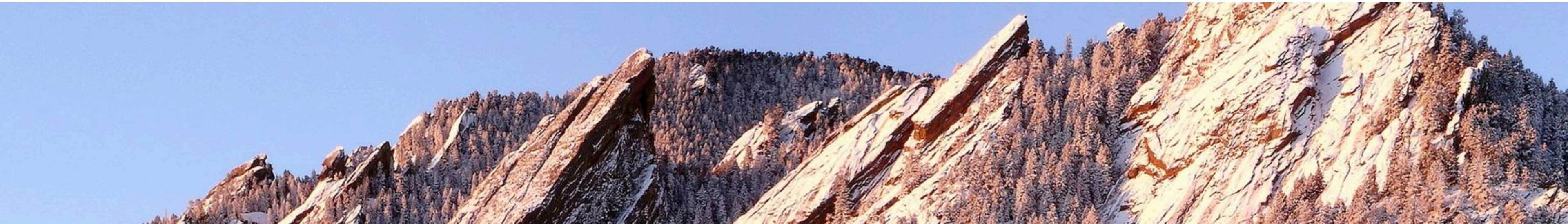
Learning Distributional Representations for Structured Output Prediction

Vivek Srikumar

Stanford University Natural Language Processing NLP

THE UNIVERSITY OF UTAH

Joint work with Chris Manning



Big picture

- Distributional representations for inputs is a success story
  - Eg. word vectors
- ***Outputs*** are discrete objects
  - One of a set of labels (document classification)
  - Label sequences (POS tagging, Chunking, NER)
  - Trees with labeled edges/nodes (Parsing)
  - Arbitrary graphs (Semantic Role Labeling, event extraction)
- Are outputs ***really*** discrete objects?

Overview

1. Background and Notation

2. Motivation
   – Why represent structures as dense vectors

1. Distributional Structured Output
   – Representing labels by real-valued vectors
   – Building structures by composing label vectors

1. Learning
   – Both label vectors and weights

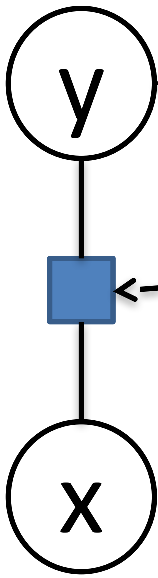
2. Experiments

Background and Notation

Notation

- Input: $\mathbf{x}$
  - Documents, sentences, paragraphs, etc
- Output: $\mathbf{y}$
  - One of multiple classes
  - Sequences
  - Trees
  - Generally combinatorial objects

Multiclass classification



Output can be one
of a set of labels

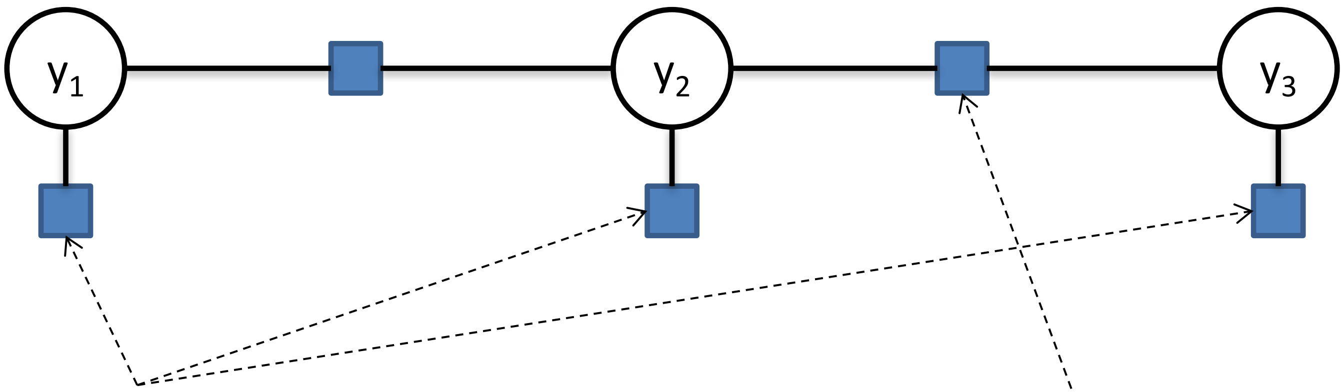
Indicates a scoring function
Typically of the form $\mathbf{w}^T\phi(\mathbf{x}, \mathbf{y})$

Weight
vector

Feature
vector

Standard definition of feature vector: Feature vector for input conjoined with label **y**

First-order sequence



Emission parts (atomic)

$$\mathbf{w}^T \phi_E(\mathbf{x}, y_i)$$

Transition parts (compositional)

$$\mathbf{w}^T \phi_T(\mathbf{x}, y_{i-1}, y_i)$$

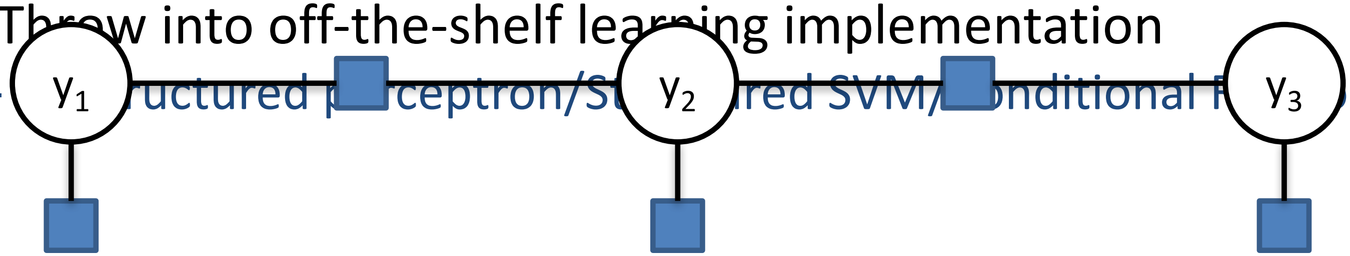
The usual approach:
For the score of a label, conjoin it with emission features to define feature vector

The usual approach:
A score for each pair of labels

That is, conjoin consecutive labels

Components of a structured model

1. Identify inputs and the output structure
   – (Sentence, POS sequence), (Document, label), etc
2. Factorize the structure
   – a.k.a define the model (linear chain, grand-parent dependencies, head-driven model, etc) What are the parts that I need to score
3. Define the scoring function
   – In NLP, nearly always linear, so effectively, "**Define the features for each factor/part**"
4. Write the inference algorithm
   – Combinatorial optimization: Dynamic programming, ILP, dual decomposition, and the like. Depends on how you do step 2
5. Throw into off-the-shelf learning implementation
   – Structured perceptron/Structured SVM/Conditional Random Field



Examples

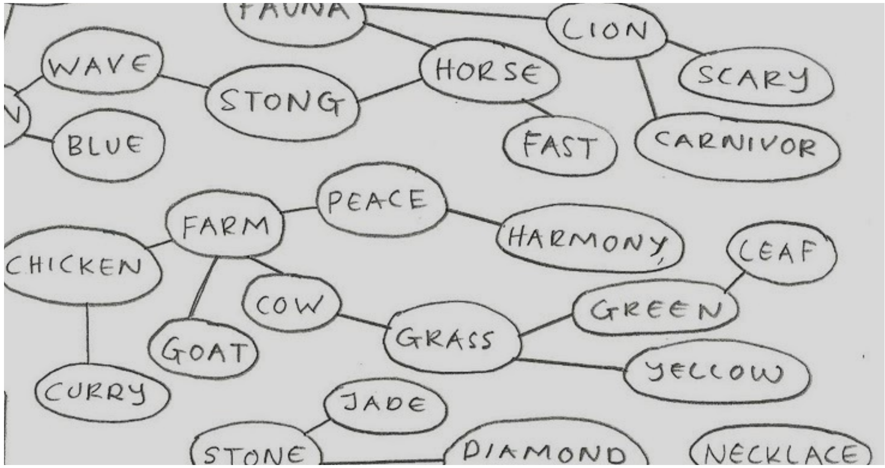
	Document classification	**First-order sequence model for POS tagging**
Input	Document	Sentence
Output	One of a set of labels	A sequence of labels
Features	Document features conjoined with each label	**Emission**: Each element of sequence conjoined with word features **Transition**: Consecutive labels conjoined
Inference	Enumerate labels	Viterbi

Are structures discrete units of meaning?

Words are not discrete units of meaning

Dense vectors allow statistical information to be shared across different words

- Example: Word vector representations
- Compare to sparse feature vectors



Are labels discrete units of meaning?


```
alt.atheism
comp.graphics
comp.os.ms-windows.misc
comp.sys.ibm.pc.hardware
comp.sys.mac.hardware
comp.windows.x
misc.forsale
rec.autos
rec.motorcycles
rec.sport.baseball
rec.sport.hockey
sci.crypt
sci.electronics
sci.med
sci.space
soc.religion.christian
talk.politics.guns
talk.politics.mideast
talk.religion.misc
```

POS tags are not discrete units of meaning

Table 2
The Penn Treebank POS tagset.

1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition/subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol (mathematical or scientific)
25.	TO	*to*
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund/present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd ps. sing. present
32.	VBZ	Verb, 3rd ps. sing. present
33.	WDT	*wh*-determiner
34.	WP	*wh*-pronoun
35.	WP$	Possessive *wh*-pronoun
36.	WRB	*wh*-adverb
37.	#	Pound sign
38.	$	Dollar sign
39.	.	Sentence-final punctuation
40.	,	Comma
41.	:	Colon, semi-colon
42.	(	Left bracket character
43.	)	Right bracket character
44.	"	Straight double quote
45.	‘	Left open single quote
46.	“	Left open double quote
47.	’	Right close single quote
48.	”	Right close double quote

Labels are not discrete units of meaning

PropBank semantic role labeling

Labels are `A0, A1, A2, A3, A4, A5,` etc.

- `A0` mostly maps to Agent
- `A1` maps to Patient or Theme (rarely also Topic)
- `A2` maps to Patient, Recipient, Experiencer, Attribute, …
- The rest don't necessarily generalize easily

Labels are ***cluster concepts*** with shared semantics

FrameNet labels have problems

– `Abuser, Assailant, Cognizer,` etc. are all agentive

Are *structures* discrete units of meaning?

POS tag sequences

- Different kinds of nouns are closer in meaning compared to verbs
- Transition from JJ -> NNS be “similar” to the transition from JJS -> NNP
  - Adjective -> Noun

So some sequences closer in meaning to each other than others

- Compare
  - DT – JJ – NN – VB – DT – JJ – NN
  - DT – JJ – NNS – VB – DT – JJR – NN
  - DT – NN– NNS – MD – VB – JJ – NN

Are *structures* discrete units of meaning?

Jack ***bought*** a glove from Mary.

Buyer	Goods	Seller	**COMMERCE_GOODS_TRANSFER**
A0	A1	A2	

Jack ***acquired*** a glove from Mary.

Recipient	Theme	Source	**ACQUIRE**
A0	A1	A2	

Jack ***returned*** a glove to Mary.

Agent	Theme	Recipient
A0	A1	A2

Distributional Structured Output

Standard models assume discrete structured output

Allocates a separate part of the parameter vector for scoring each label

- Also for compositions of labels (i.e. like transitions in a sequence model)

Ignores the fact that information can be shared across vaguely defined labels

What we want

1. Represent atomic labels to allow sharing of statistical information across them

1. Define an operator that allows us to construct compositional structures from atomic labels

2. Use all the advances in structured prediction (eg. inference)

Components of a structured model

1. Identify inputs and the output structure
   – (Sentence, POS sequence), (Document, label), etc
2. Factorize the structure
   – a.k.a define the model (linear chain, grand-parent dependencies, head-driven model, etc) What are the parts that I need to score
3. Define the scoring function
   – In NLP, nearly always linear, so effectively, "**Define the features for each factor/part**"
4. Write the inference algorithm
   – Combinatorial optimization: Dynamic programming, ILP, dual decomposition, and the like. Depends on how you do step 2
5. Throw into off-the-shelf learning implementation
   – Structured perceptron/Structured SVM/Conditional Random Field

Atomic Labels: Multiclass classification

Inputs to be classified $\mathbf{x} \in \Re^n$

Labels $y \in \{1, 2, 3, \cdots, m\}$

Weight vector $\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_m \end{bmatrix} \in \Re^{mn}$

$$\text{Prediction}(\mathbf{x}) = \boxed{\text{Highest scoring label}}$$

$$= arg \max_y \mathbf{w}^T \phi(\mathbf{x}, y)$$

Feature vector for assigning label y to x

$$\phi(\mathbf{x}, y) = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{x} \\ \vdots \\ \mathbf{0} \end{bmatrix}_{mn \times 1}$$

The y^{th} block

1. Notational convenience
2. Can be extended to structured y's

Each label y allocated a different part of the weight space

$arg\max_y \mathbf{w}^T \phi(\mathbf{x}, y)$

Let's examine the features ϕ

Feature vector for assigning label y to x

$$\phi(\mathbf{x}, y) = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{x} \\ \vdots \\ \mathbf{0} \end{bmatrix}_{mn \times 1}$$

The y^{th} block

$$A_y = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \in \Re^m$$

The y^{th} element

Define a **one-hot** vector A_y for each label y

We have $\mathbf{x} A_y^T = [\mathbf{0} \cdots \mathbf{x} \cdots \mathbf{0}]_{n \times m}$

We can rewrite the feature vector in terms of the label vectors

$$\phi(\mathbf{x}, y) = vec(\mathbf{x} A_y^T) = vec(\mathbf{x} \otimes A_y)$$

Vec vectorizes its argument columnwise

Recall, the usual method calls for a conjunction of x with label y

A reformulation of multi-class classification

$$\mathbf{w}^T\phi(\mathbf{x},\mathbf{y}) = \mathbf{w}^T vec(\mathbf{x}\otimes A_y)$$

Inputs to be classified $\mathbf{x}\in\Re^n$

Labels $y\in\{1,2,3,\cdots,m\}$

If the label vectors A_y are one-hot vectors, under this scheme, we recover standard multi-class classification

- No sharing of information across labels
- Each label accesses a different part of **w**

What if they weren't one-hot?

Make the label vectors real valued

With one-hot label vectors:

- Each label associated with separate region of weight space
- Label semantics is ignored

Label vectors do not need to be one-hot

- Share weights across labels via the redefinition of features

What we want

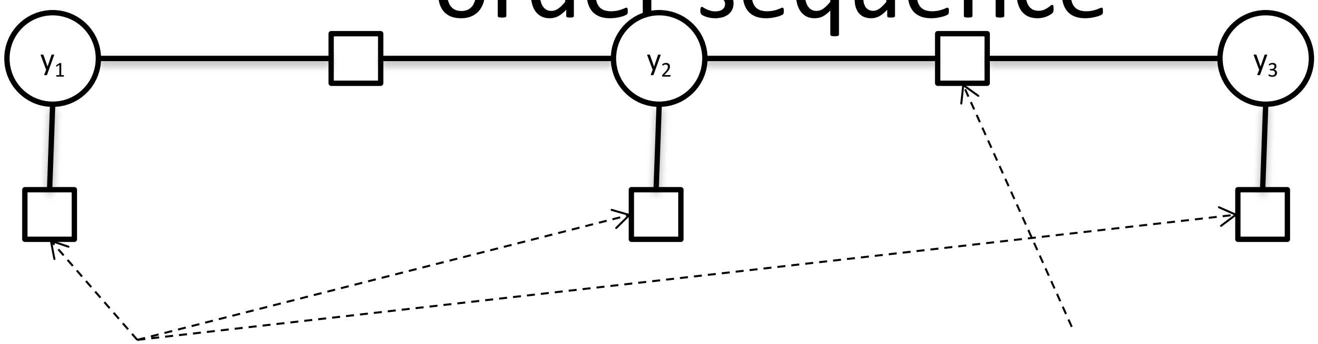
1. Represent atomic labels to allow sharing of statistical information across them

$$\boxed{\text{Represent labels as dense vectors and define feature vectors appropriately}}$$

1. Define an operator that allows us to construct compositional structures from atomic labels

2. Use all the advances in structured prediction (eg. inference)

Compositional Structure: A first-order sequence



Emission parts (atomic)

i^{th} label conjoined with emission features to get the emission score

$$\mathbf{w}^T \phi_E(\mathbf{x}, y_i)$$

Transition parts (compositional)

Transition scores are associated with conjunctions of consecutive labels

$$\mathbf{w}^T \phi_T(\mathbf{x}, y_{i-1}, y_i)$$

As before

Let A_{y_i} represent a one-hot vector for label y_i

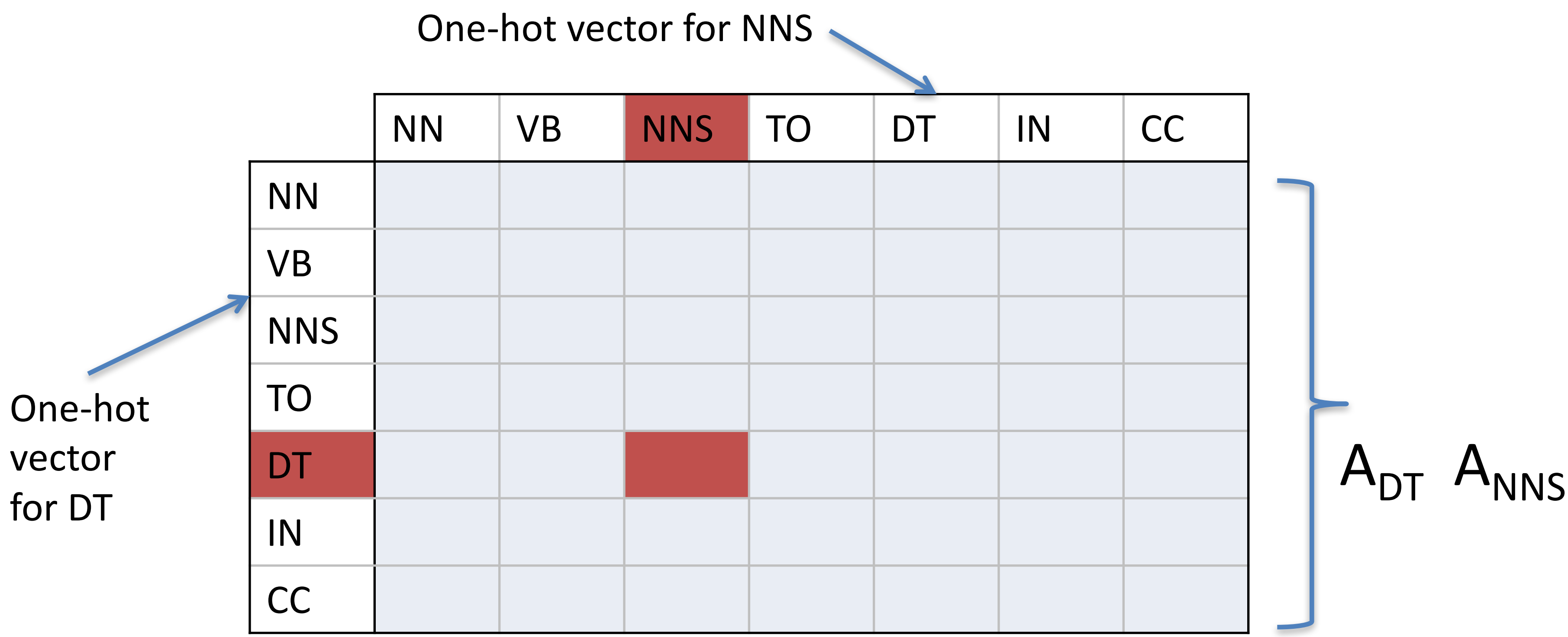
For parts involving multiple labels

$$\mathbf{w}^T vec\left(\phi_E(\mathbf{x}) \otimes A_{y_i}\right)$$

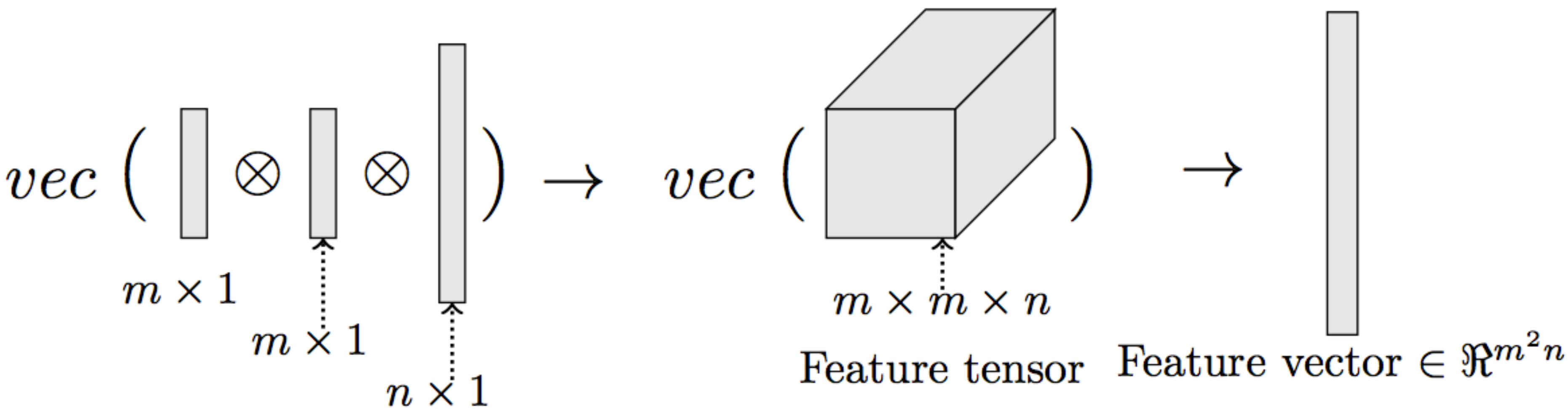
$$\mathbf{w}^T vec\left(\phi_T(\mathbf{x}) \otimes A_{y_{i-1}} \otimes A_{y_i}\right)$$

Because each label vector is one hot, we get back the standard linear model

An example of the transition outer product



Building the feature vector



Definition extends to more complex structures

For atomic parts (like multiclass): single tensor product

Eg: $\phi(\text{doc}, \texttt{alt.atheism}) = A_{\texttt{alt.atheism}}\ \phi(\text{doc})$

For compositional parts (like transitions): As many additional tensor products as compositions

$\phi(x, \texttt{NNS} \rightarrow \texttt{NNS}) = A_{\texttt{DT}}\ A_{\texttt{NNS}}\ \phi(x)$

$\phi(x, [\texttt{DT,NNS}] \rightarrow \texttt{NNS}) = A_{\texttt{DT}}\ A_{\texttt{NNS}}\ A_{\texttt{NNS}}\ \phi(x)$

$\phi(x, \texttt{return} \rightarrow \texttt{A0}) = A_{\texttt{return}}\ A_{\texttt{A0}}\ \phi(x)$

….

*DI*stributional *ST*ructured *O*utput (DISTRO)

1. Represent atomic labels to allow sharing of statistical information across them

Represent labels as dense vectors and define feature vectors appropriately

1. Define an operator that allows us to construct compositional structures from atomic labels

Represent label compositions using tensor products to extend feature vectors for larger parts/structures

2. Uses all the advances in structured prediction (eg. inference)

Related ideas

- Two layer neural network for multi-class classification
  - Effectively learn a representation for each label
- Nati Srebro's thesis for multi-class classification
  - Looks at this as matrix factorization
- Paul Smolensky (1990), Geoff Hinton 1986, Tony Plate thesis (mid-90's)

Learning

- We still need to learn the weight vector
- Also, for each label, we learn a vector representation
- Need to specify the dimensionality of the vector
  - Side note: if dimensionality = number of labels, we might as well use one-hot vectors

The learning setting

Input: A dataset (x,y)

- y is a structure composed of discrete labels

Need a to specify the shape of the structure as with any CRF/Struct SVM

- Linear sequence model for POS
- Tree for a dependency parse

Need to specify inference algorithm

That is, same setting as standard structured learning

The learning objective

Label vectors specify the feature representation of a structure

Learn by minimizing

Nuclear norm, favors lower rank matrices

$$\boxed{\text{Weight regularization}} + \boxed{\text{Label vector regularization}}$$

$$+ \boxed{\text{Loss on the training data as a function of w and A}}$$

Hinge loss here for Structured SVM,
Could be changed to get CRF

w: Weight vector
A: Matrix composed of label vectors as columns

Learning weights and label vectors

Objective is non convex

(For multiclass classification, bilinear in w and A)

- An alternating algorithm
  1. Initialize w and A
  2. Iterate
     1. Fix A, solve for w using SGD
     2. Fix w, solve for A using SGD with proximal step

Experiments

Dense labels help document classification

20 newsgroups classification

Setting	Accuracy
Structured SVM (one-hot vectors)	81.4
DISTRO	**84.0**
DISTRO (15 dim vectors)	83.1
DISTRO (11 dim vectors)	80.9

- 11-19 dimensional vectors generally good
- Not sensitive to initialization because loss driven

Final label similarities

Label	Id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
alt.atheism	1	1.000	0.369	-0.023	-0.173	-0.037	0.336	-0.052	0.365	-0.240	-0.051	0.253	-0.081	-0.032	0.000	-0.044	-0.054	-0.020	0.353	-0.054	-0.211
talk.politics.guns	2	0.369	1.000	-0.075	-0.065	-0.108	0.360	-0.060	-0.006	-0.070	-0.091	0.251	0.015	-0.086	-0.034	-0.074	-0.097	-0.057	0.343	-0.124	-0.047
sci.crypt	3	-0.023	-0.075	1.000	0.355	0.332	-0.040	-0.154	-0.012	-0.063	0.352	-0.055	-0.043	-0.169	-0.151	-0.075	-0.089	-0.126	-0.063	-0.107	-0.087
sci.space	4	-0.173	-0.065	0.355	1.000	0.336	-0.012	-0.120	0.002	-0.098	0.336	-0.026	-0.116	-0.143	-0.045	-0.079	-0.101	-0.116	-0.069	-0.127	-0.129
sci.med	5	-0.037	-0.108	0.332	0.336	1.000	-0.027	-0.197	-0.005	-0.119	0.312	-0.070	-0.051	-0.152	-0.063	-0.130	-0.124	-0.129	-0.041	-0.139	-0.097
talk.politics.mideast	6	0.336	0.360	-0.040	-0.012	-0.027	1.000	-0.077	-0.047	-0.043	-0.101	0.303	-0.054	-0.094	-0.057	-0.044	-0.057	-0.072	0.368	-0.039	-0.024
comp.sys.ibm.pc.hardware	7	-0.052	-0.060	-0.154	-0.120	-0.197	-0.077	1.000	-0.019	-0.134	-0.236	-0.074	-0.159	-0.051	-0.086	-0.120	-0.146	0.169	-0.057	-0.001	-0.059
soc.religion.christian	8	0.365	-0.006	-0.012	0.002	-0.005	-0.047	-0.019	1.000	0.016	-0.156	0.309	-0.171	-0.171	-0.250	0.024	-0.029	0.014	-0.055	-0.020	0.054
rec.sport.baseball	9	-0.240	-0.070	-0.063	-0.098	-0.119	-0.043	-0.134	0.016	1.000	-0.098	-0.082	-0.067	-0.080	0.260	0.212	0.308	-0.120	-0.044	-0.127	-0.236
sci.electronics	10	-0.051	-0.091	0.352	0.336	0.312	-0.101	-0.236	-0.156	-0.098	1.000	-0.048	-0.111	-0.195	-0.109	-0.106	-0.129	-0.203	-0.074	-0.183	-0.156
talk.religion.misc	11	0.253	0.251	-0.055	-0.026	-0.070	0.303	-0.074	0.309	-0.082	-0.048	1.000	0.003	-0.064	-0.007	-0.152	-0.078	-0.272	0.332	-0.111	-0.036
misc.forsale	12	-0.081	0.015	-0.043	-0.116	-0.051	-0.054	-0.159	-0.171	-0.067	-0.111	0.003	1.000	-0.156	-0.103	-0.102	-0.150	-0.180	-0.099	-0.131	-0.102
comp.graphics	13	-0.032	-0.086	-0.169	-0.143	-0.152	-0.094	-0.051	-0.171	-0.080	-0.195	-0.064	-0.156	1.000	-0.201	-0.079	-0.120	-0.033	-0.062	0.195	-0.077
rec.sport.hockey	14	0.000	-0.034	-0.151	-0.045	-0.063	-0.057	-0.086	-0.250	0.260	-0.109	-0.007	-0.103	-0.201	1.000	0.365	0.288	-0.123	-0.060	-0.064	-0.061
rec.motorcycles	15	-0.044	-0.074	-0.075	-0.079	-0.130	-0.044	-0.120	0.024	0.212	-0.106	-0.152	-0.102	-0.079	0.365	1.000	0.283	-0.126	-0.032	-0.130	-0.103
rec.autos	16	-0.054	-0.097	-0.089	-0.101	-0.124	-0.057	-0.146	-0.029	0.308	-0.129	-0.078	-0.150	-0.120	0.288	0.283	1.000	-0.154	-0.047	-0.117	-0.111
comp.sys.mac.hardware	17	-0.020	-0.057	-0.126	-0.116	-0.129	-0.072	0.169	0.014	-0.120	-0.203	-0.272	-0.180	-0.033	-0.123	-0.126	-0.154	1.000	-0.130	0.022	-0.004
talk.politics.misc	18	0.353	0.343	-0.063	-0.069	-0.041	0.368	-0.057	-0.055	-0.044	-0.074	0.332	-0.099	-0.062	-0.060	-0.032	-0.047	-0.130	1.000	-0.049	-0.016
comp.windows.x	19	-0.054	-0.124	-0.107	-0.127	-0.139	-0.039	-0.001	-0.020	-0.127	-0.183	-0.111	-0.131	0.195	-0.064	-0.130	-0.117	0.022	-0.049	1.000	-0.038
comp.os.ms-windows.misc	20	-0.211	-0.047	-0.087	-0.129	-0.097	-0.024	-0.059	0.054	-0.236	-0.156	-0.036	-0.102	-0.077	-0.061	-0.103	-0.111	-0.004	-0.016	-0.038	1.000

Compositional structure: POS tagging

- English: Penn Treebank, 45 labels

Setting	Accuracy
Structured SVM (45 dim one-hot vectors)	96.2
DISTRO (5 dim vectors)	95.1
DISTRO (20 dim vectors)	**96.7**

- Basque: CoNLL 2007 shared task data, 64 labels

Setting	Accuracy
Structured SVM (64 dim one-hot vectors)	91.5
DISTRO	**92.4**

Summary

- Structures aren’t always discrete units of meaning
  - Dense representations
- A method for using dense vector representations of labels for arbitrary structured prediction
- An objective for learning label vectors and weights together