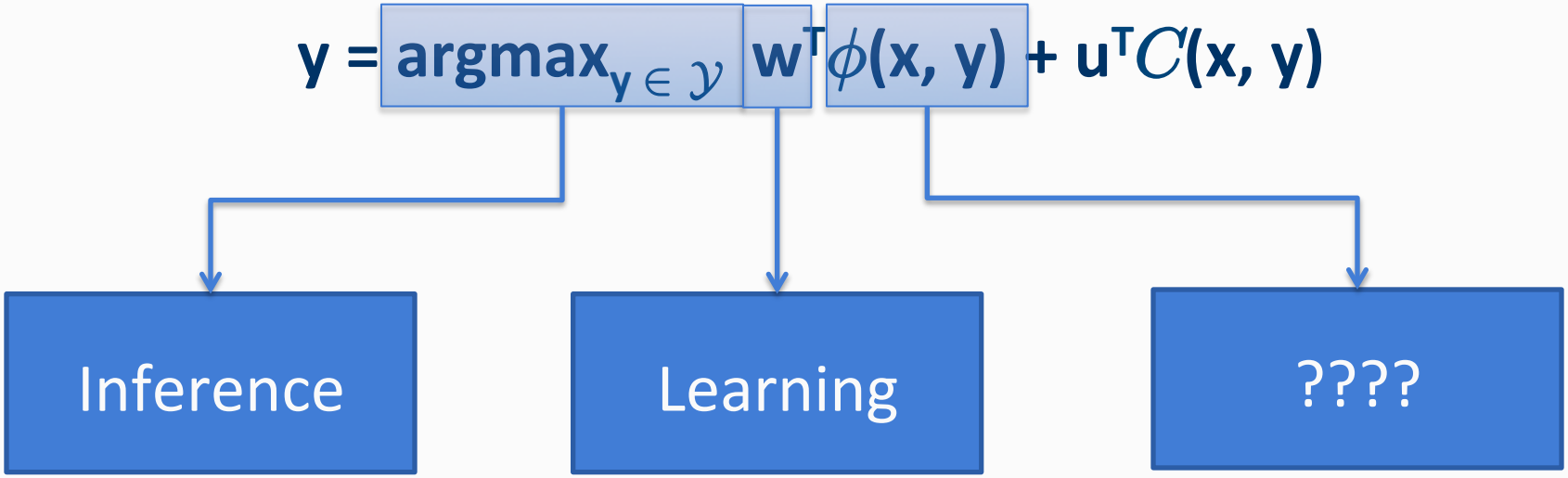
Part 4: Learning Distributed Representations for Structured Output Prediction

[Srikumar & Manning, NIPS 2014]

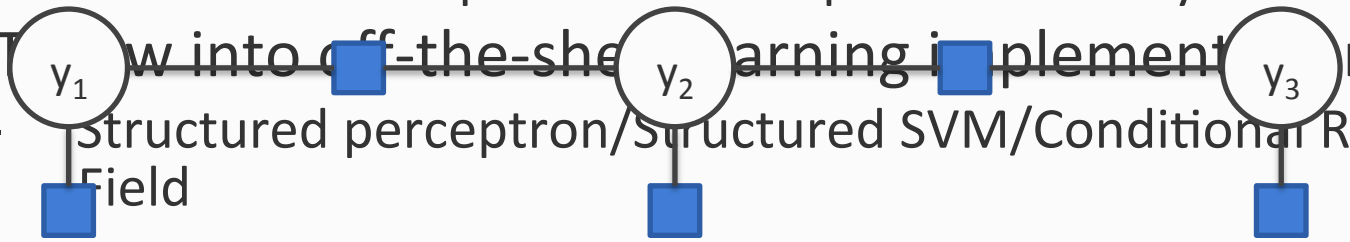
Let's look at the prediction step

Can we say something about the features?



Components of a structured model

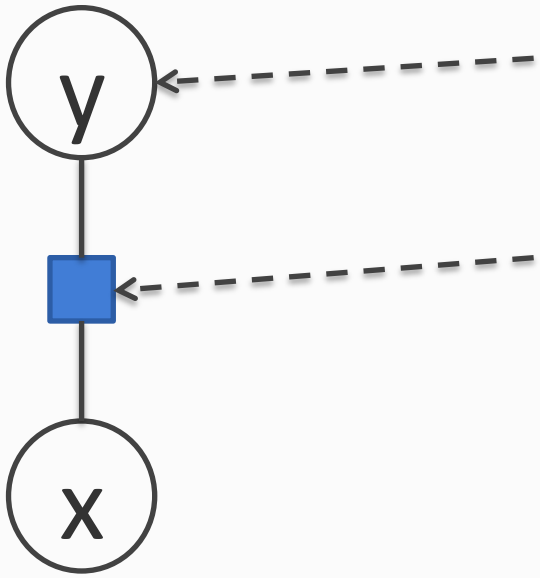
1. Identify inputs and the output structure
   - (Sentence, POS sequence), (Document, label), etc
2. Factorize the structure
   - a.k.a define the model (linear chain, grand-parent dependencies, head-driven model, etc) "What are the parts that I need to score?"
3. Define the scoring function
   - With linear models, **define the features for each factor/part**
4. Write the inference algorithm
   - Combinatorial optimization: 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

Multiclass classification



Output can be one of a set of labels

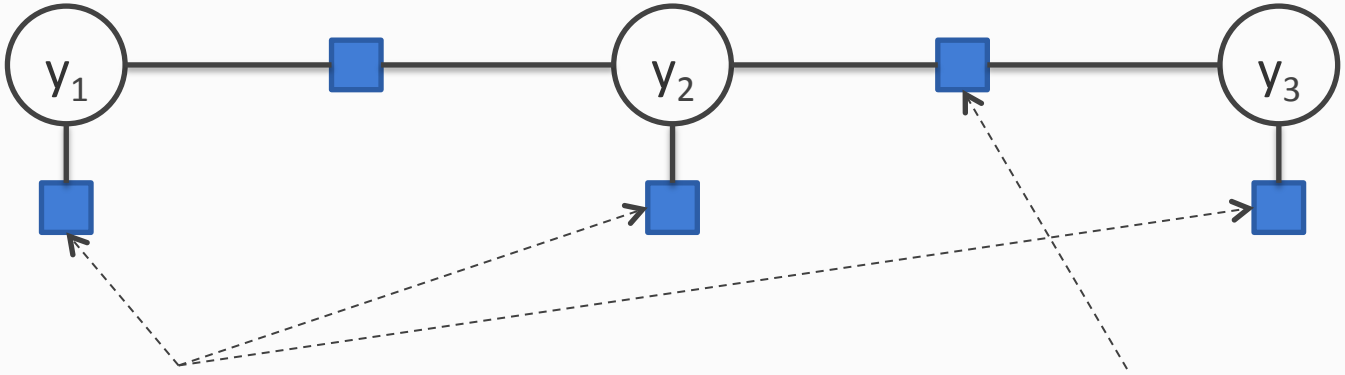
Indicates a scoring function
Typically of the form $\mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Weight vector

Feature vector

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

First-order sequence



Emission parts (atomic)

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

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

Transition parts (compositional)

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

The usual approach:
A score for each pair of labels

That is, conjoin consecutive labels

Distributed representations: Big picture

"…a distributed representation occurs when … meaning is represented by a pattern of activity across a number of processing units" [Hinton et al. 1986]

- Distributed representations for inputs: A success story
  - E.g. word vectors

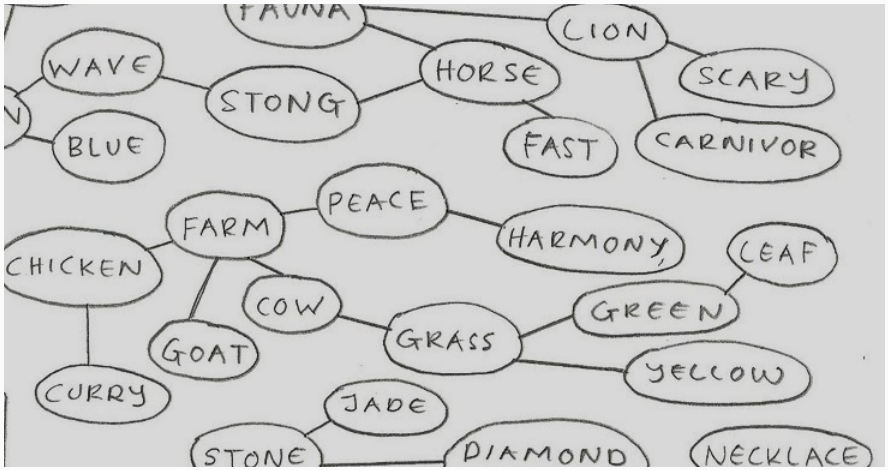
- ***Outputs*** are discrete objects
  - One of a set of labels (document classification)
  - Label sequences (POS tagging)
  - Trees with labeled edges/nodes (Parsing)
  - Arbitrary graphs (Semantic Role Labeling, event extraction)

- Can we think of distributed representations for structures?
  - Starting with individual labels to compose full structures

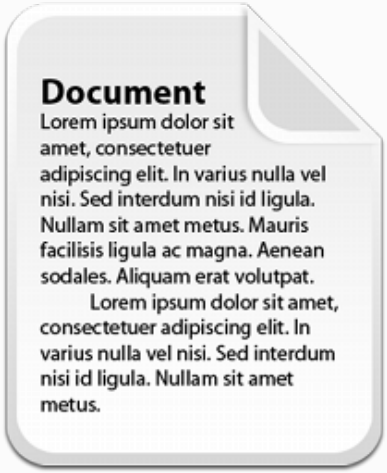
Words are not discrete units of meaning

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

Are *structures* discrete units of meaning?

POS tag sequences

- Different kinds of nouns are closer in meaning compared to verbs
- Transition JJ $\rightarrow$ NNS "similar" to the transition JJS $\rightarrow$ NNP
  - Both Adjective $\longrightarrow$ Noun

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.

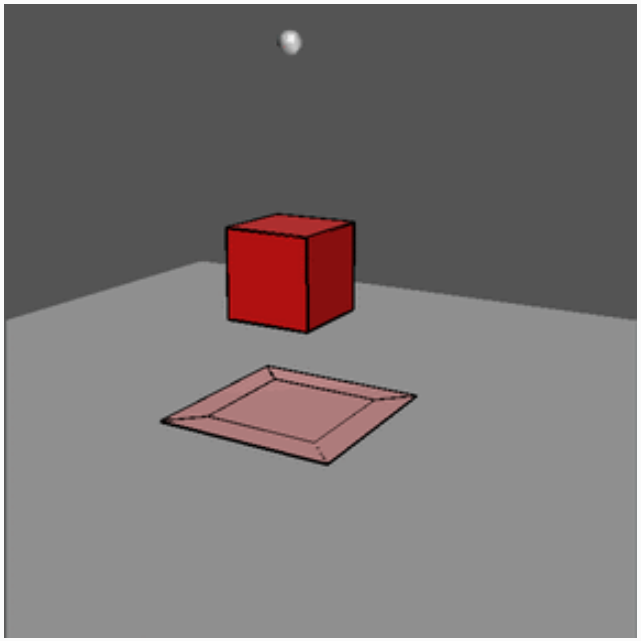
Buyer **Goods** **Seller** **COMMERCE_GOODS_TRANSFER**

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

Recipient **Theme** **Source** **ACQUIRE**

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

Agent **Theme** **Recipient**



Distributed Structured Output

a.k.a Embedding structured outputs

[Image courtesy Davide P. Cervone]

Standard models assume discrete outputs

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 labels that are "*cluster concepts*"

What we want

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

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

3. 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
   - With linear models, **define the features for each factor/part**
4. Write the inference algorithm
   - Combinatorial optimization: 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 $\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_m \end{bmatrix} \in \Re^{mn}$

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

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

Kesler Construction

Feature vector for assigning label y to **x**

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

The y^{th} block

$mn \times 1$

Notational convenience

Extends to structured **y**'s

Each label y allocated a different part of the weight space

[Zimak, Har-Peled & Roth, NIPS 2002]

Let's examine the features ϕ

Kesler Construction

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

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

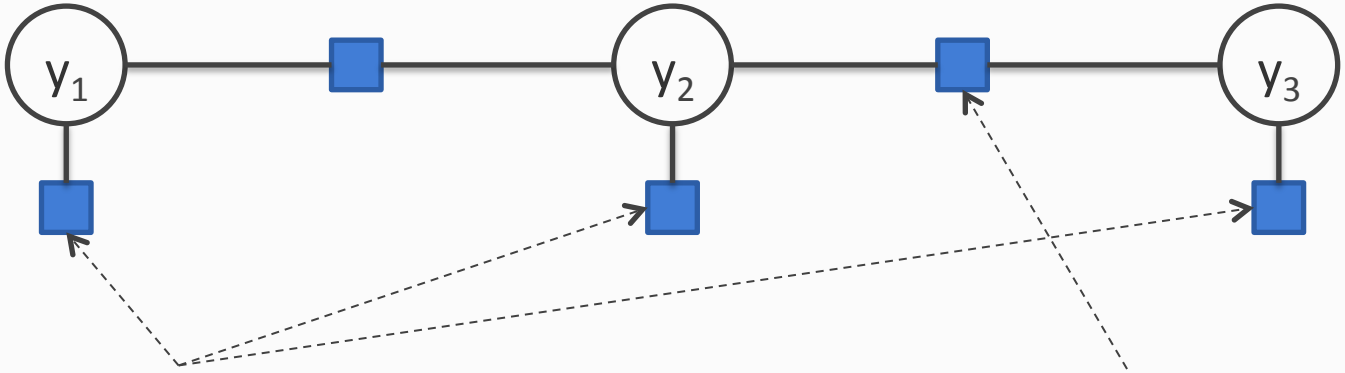
   Represent labels as dense vectors and define feature vectors appropriately

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

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

Compositional Structure: First-order sequence

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



Emission parts (atomic)

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

As before

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

Transition parts (compositional)

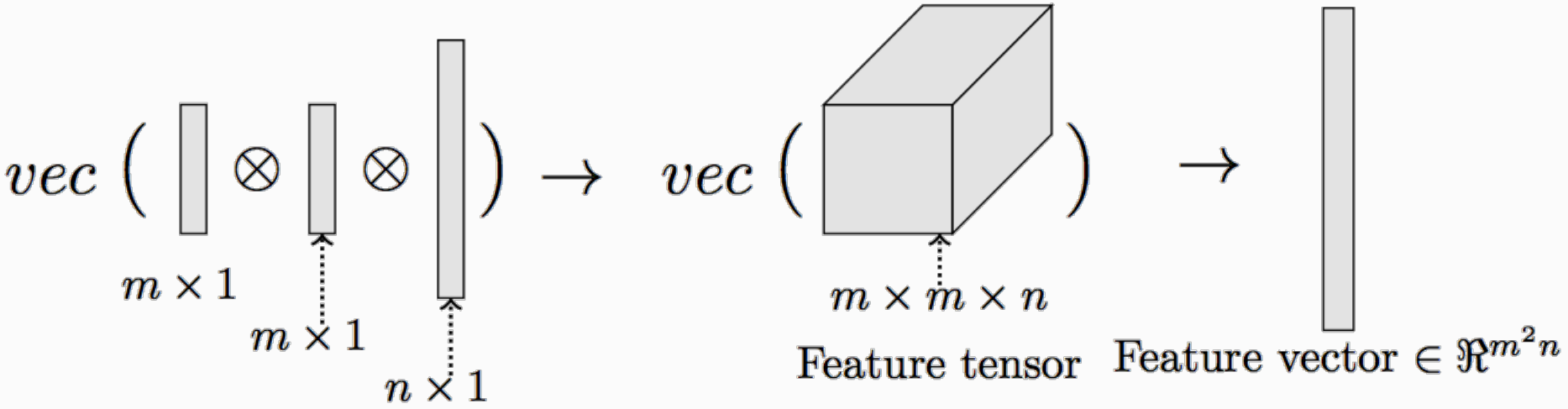
$$\mathbf{w}^T \phi_T(\mathbf{x}, y_{i-1}, y_i)$$

For parts involving multiple labels

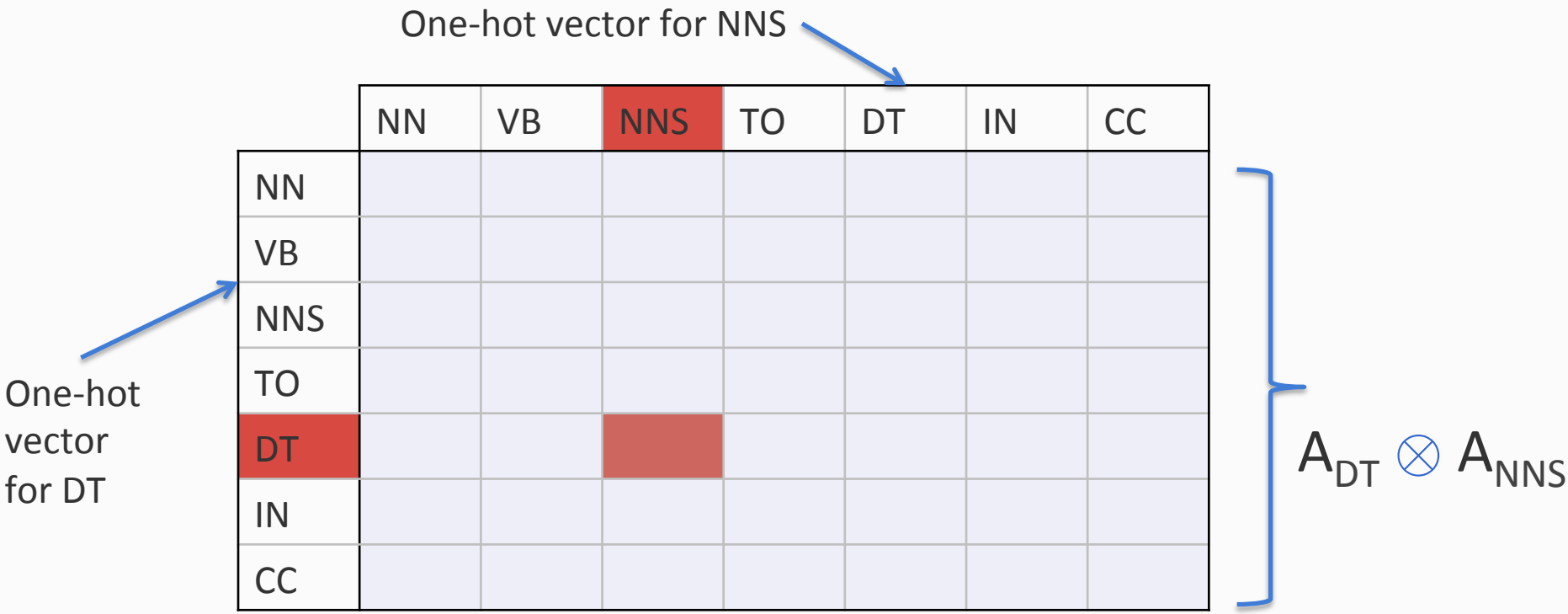
$$\mathbf{w}^T vec\left(\phi_T(\mathbf{x}) \otimes A_{y_{i-1}} \otimes A_{y_i}\right)$$

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

Building the feature vector



An example of the transition outer product



Definition extends to more complex structures

For atomic parts (like multiclass): single tensor product

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

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

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

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

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

....

*DI*stributed *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

2. 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

3. 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

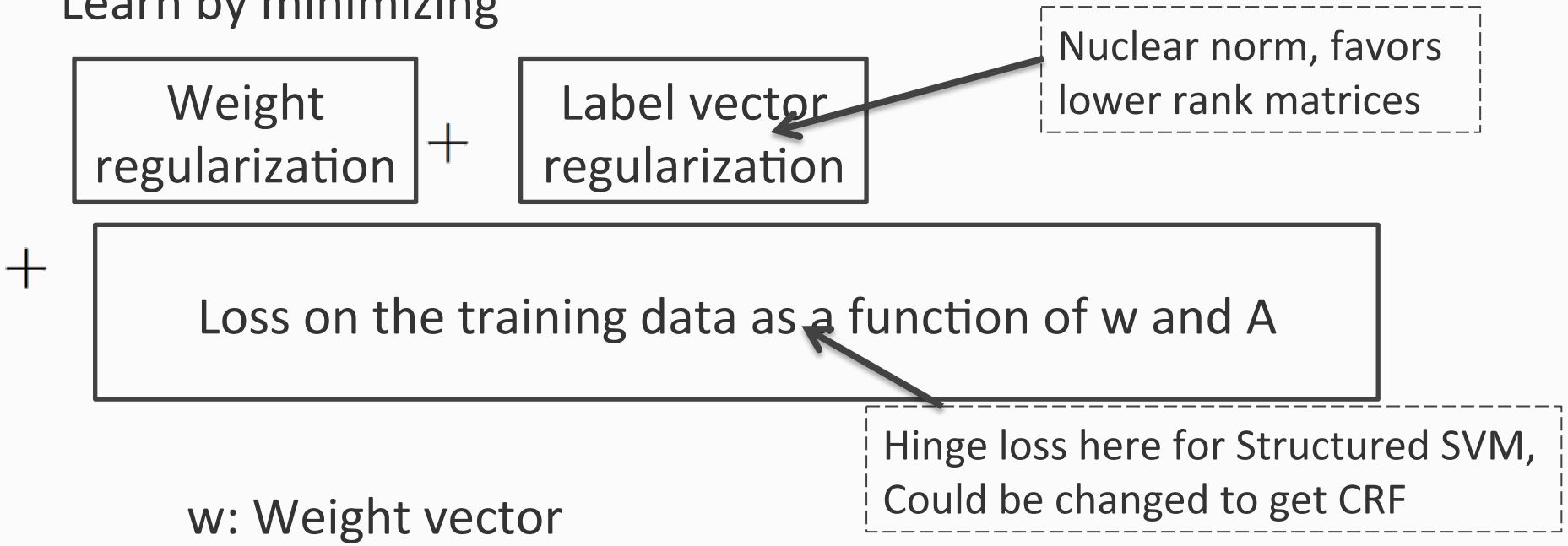
- Linear sequence model for POS
- Tree for a dependency parse

That is, same setting as standard structured learning

The learning objective

Label vectors specify the feature representation of a structure

Learn by minimizing



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)

Alternating minimization 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

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