General Formulations for Structures: Constrained Conditional Models

CS 6355: Structured Prediction

Where are we?

- Graphical models
  - Bayesian Networks
  - Markov Random Fields
- Formulations of structured output
  - Joint models
    - Markov Logic Network
  - Conditional models
    - Conditional Random Fields (again)
    - Constrained Conditional Models

Outline

- Consistency of outputs and the value of inference
- Constrained conditional models via an example
- Hard constraints and Integer Programs
- Soft constraints

Outline

- Consistency of outputs and the value of inference
- Constrained conditional models via an example
- Hard constraints and Integer Programs
- Soft constraints

Consistency of outputs

Or: How to introduce knowledge into prediction

Suppose we have a sequence labeling problem where the outputs can be one of A or B

Consistency of outputs

Or: How to introduce knowledge into prediction

Suppose we have a sequence labeling problem where the outputs can be one of A or B

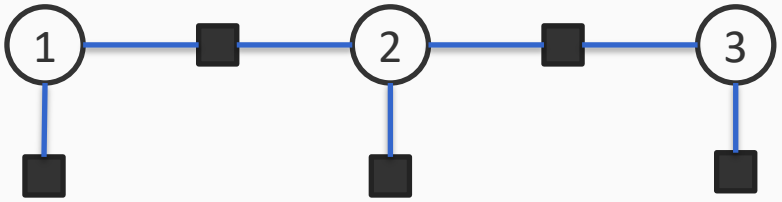
Suppose, for some example, we have a sequence with three steps

Consistency of outputs

Or: How to introduce knowledge into prediction

Suppose we have a sequence labeling problem where the outputs can be one of A or B

Suppose, for some example, we have a sequence with three steps
We can define a linear chain factor graph with unary and binary terms

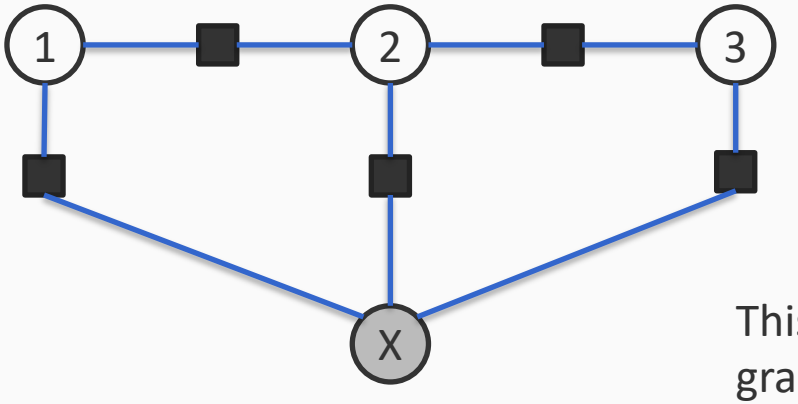


Consistency of outputs

Or: How to introduce knowledge into prediction

Suppose we have a sequence labeling problem where the outputs can be one of A or B

Suppose, for some example, we have a sequence with three steps
We can define a linear chain factor graph with unary and binary terms

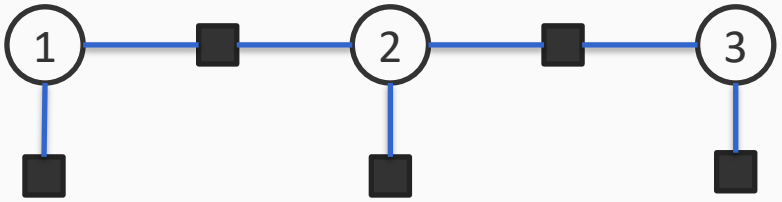


This is short hand for this factor graph. Going ahead, we will not show the inputs. We will assume that we have conditional models.

Consistency of outputs

Or: How to introduce knowledge into prediction

Suppose we have a sequence labeling problem where the outputs can be one of A or B



Possible outputs

AAA
AAB
ABA
ABB
BAA
BAB
BBA
BBB

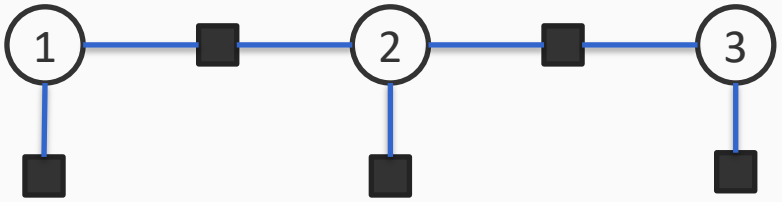
Consistency of outputs

Or: How to introduce knowledge into prediction

Suppose we have a sequence labeling problem where the outputs can be one of A or B

We want to add a condition:

There should be no more than one B in the output



Possible outputs

AAA
AAB
ABA
ABB
BAA
BAB
BBA
BBB

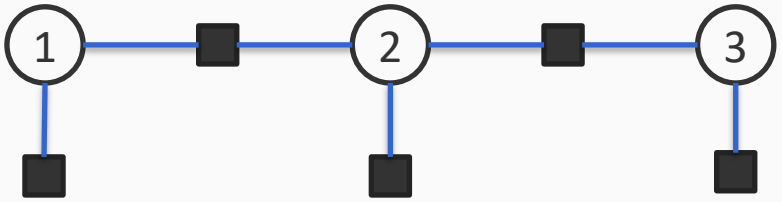
Consistency of outputs

Or: How to introduce knowledge into prediction

Suppose we have a sequence labeling problem where the outputs can be one of A or B

We want to add a condition:

There should be no more than one B in the output



Possible outputs

AAA
AAB
ABA
~~ABB~~
BAA
~~BAB~~
~~BBA~~
~~BBB~~

Consistency of outputs

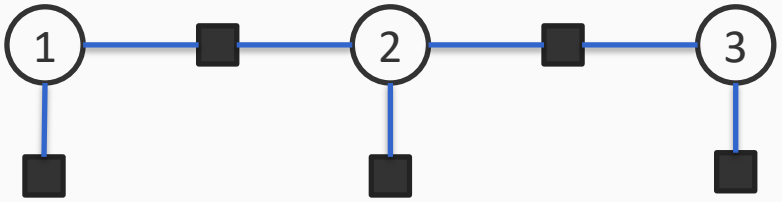
Or: How to introduce knowledge into prediction

Suppose we have a sequence labeling problem where the outputs can be one of A or B

We want to add a condition:

There should be no more than one B in the output

How can we do that? Which decisions interact in this condition?



Possible outputs

AAA
AAB
ABA
~~ABB~~
BAA
~~BAB~~
~~BBA~~
~~BBB~~

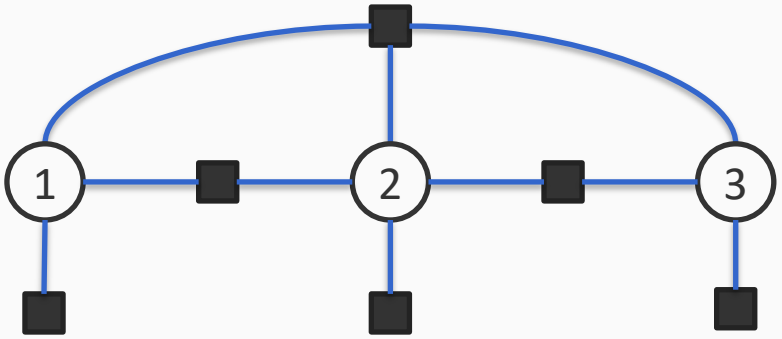
Consistency of outputs

Or: How to introduce knowledge into prediction

Suppose we have a sequence labeling problem where the outputs can be one of A or B

We want to add a condition:

There should be no more than one B in the output



Possible outputs

AAA

AAB

ABA

~~ABB~~

BAA

~~BAB~~

~~BBA~~

~~BBB~~

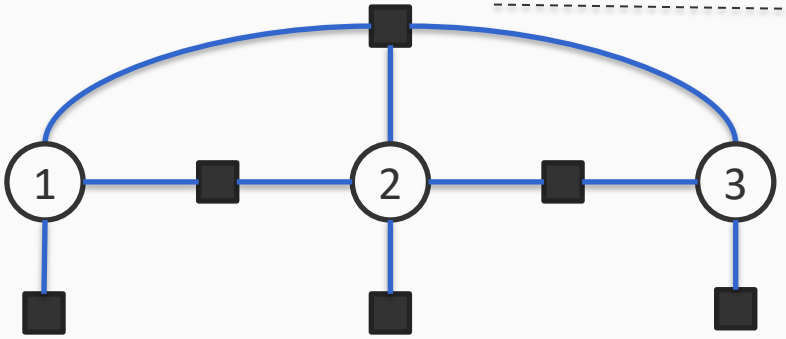
Consistency of outputs

Or: How to introduce knowledge into prediction

Suppose we have a sequence labeling problem where the outputs can be one of A or B

We want to add a condition:

There should be no more than one B in the output



This potential function ensures the condition

y1	y2	y3	f
A	A	A	1
A	A	B	1
A	B	A	1
A	B	B	0
B	A	A	1
B	A	B	0
B	B	A	0
B	B	B	0

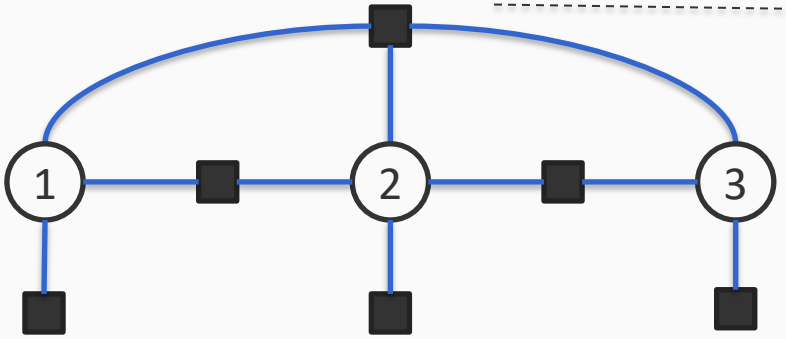
Consistency of outputs

Or: How to introduce knowledge into prediction

Suppose we have a sequence labeling problem where the outputs can be one of A or B

We want to add a condition:

There should be no more than one B in the output



This potential function ensures the condition

y1	y2	y3	f
A	A	A	1
A	A	B	1
A	B	A	1
A	B	B	0
B	A	A	1
B	A	B	0
B	B	A	0
B	B	B	0

But the standard CRF learning does not allow for potential functions to be set manually

Should we learn what we can write down easily?

Especially for such large, computationally cumbersome factors

Another look at learning and inference

- **Inference**: A global decision comprising of multiple local decisions and their inter-dependencies

  If there were no inter-dependencies between decisions, we could as well treat these as independent prediction problems

Another look at learning and inference

- **Inference**: A global decision comprising of multiple local decisions and their inter-dependencies

  If there were no inter-dependencies between decisions, we could as well treat these as independent prediction problems

- Does **learning** need to be global too?

  Recall: Local vs. global learning

  - Global learning: Learn with inference
  - Local learning: Learn the local decisions independently and piece them together
    - Maybe we can learn *sub-structures* independently and piece them together

Typical updates in global learning

Stochastic gradient descent update for CRF

- For a training example $(\mathbf{x}_i, \mathbf{y}_i)$:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha_t \left(\Phi(\mathbf{x}_i, \mathbf{y}_i) - E_{\mathbf{y} \sim P(\cdot|\mathbf{x}_i, \mathbf{w})} [\Phi(\mathbf{x_i}, \mathbf{y}_i)] \right)$$

Typical updates in global learning

Stochastic gradient descent update for CRF

- For a training example $(\mathbf{x}_i, \mathbf{y}_i)$:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha_t\left(\Phi(\mathbf{x}_i, \mathbf{y}_i) - E_{\mathbf{y}\sim P(\cdot|\mathbf{x}_i,\mathbf{w})}[\Phi(\mathbf{x_i}, \mathbf{y}_i)]\right)$$

Structured perceptron

- For a training example $(\mathbf{x}_i, \mathbf{y}_i)$:

$$\hat{y} = \max_{\mathbf{y}} \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y})$$

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha_t(\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \hat{\mathbf{y}})$$

Typical updates in global learning

Stochastic gradient descent update for CRF

- For a training example $(\mathbf{x}_i, \mathbf{y}_i)$:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha_t \left(\Phi(\mathbf{x}_i, \mathbf{y}_i) - E_{\mathbf{y} \sim P(\cdot|\mathbf{x}_i, \mathbf{w})} [\Phi(\mathbf{x}_i, \mathbf{y}_i)] \right)$$

Structured perceptron

- For a training example $(\mathbf{x}_i, \mathbf{y}_i)$:

$$\hat{y} = \max_{\mathbf{y}} \mathbf{w}^T \phi(\mathbf{x}_i, \mathbf{y})$$

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha_t (\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \hat{\mathbf{y}})$$

Some form of inference needed within the innermost loop

Why local learning?

- Global learning may not always be feasible
  - Practical concerns

    Global learning can be computationally prohibitive because of inference within the training loop

  - Data issues
    - What if we don't have a single dataset that is fully annotated with a structure
    - Instead, we have multiple datasets, each with a subset of the structures

- But inference *at deployment* can still be global
  - Recall the discussion of local vs global learning from multiclass classification

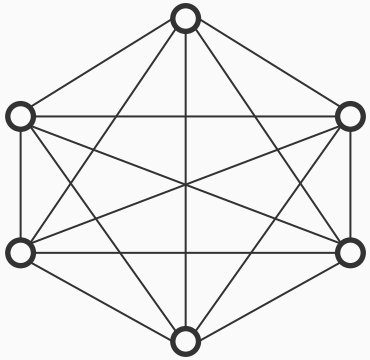
    In the one-vs-all case, learning was local, but prediction was not.

Questions?

Combining local classifiers

Inference can "glue" together local decisions

- And enforce *global* coherence



3 possible node labels 

3 possible edge labels 

Setting

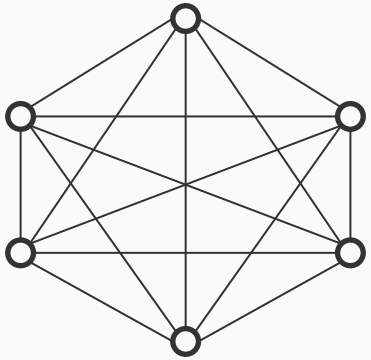
Output: Nodes and edges are labeled and the blue and orange edges form a tree

Goal: Find the highest scoring tree

Combining local classifiers

Inference can "glue" together local decisions

– And enforce *global* coherence

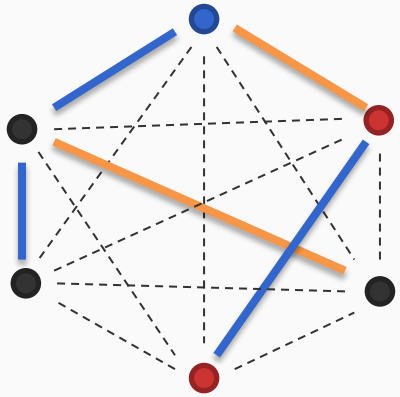


3 possible node labels

3 possible edge labels

$$\arg\max_{\mathbf{y}} \quad \text{score}(\mathbf{x}, \mathbf{y})$$
$$\text{s.t.} \quad \mathbf{y} \text{ forms a tree}$$

We need to ensure that the colored edges form a valid output (i.e. a tree)



Setting

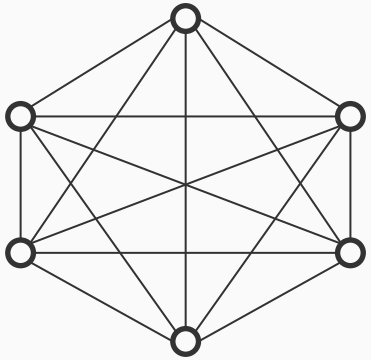
Output: Nodes and edges are labeled and the blue and orange edges form a tree

Goal: Find the highest scoring tree

Combining local classifiers

Inference can "glue" together local decisions

– And enforce *global* coherence



3 possible node labels

3 possible edge labels

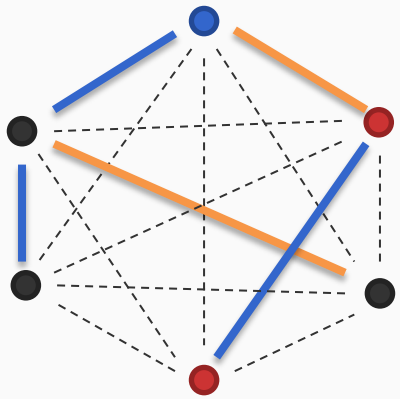
Setting

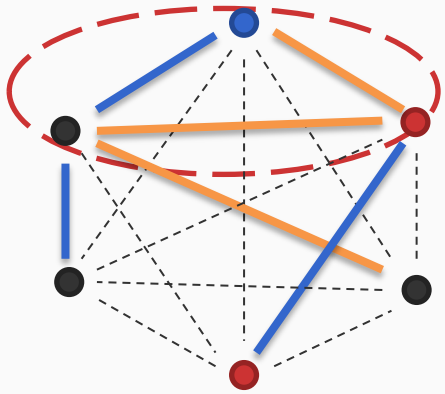
Output: Nodes and edges are labeled and the blue and orange edges form a tree

Goal: Find the highest scoring tree

$$\arg\max_{\mathbf{y}} \quad \text{score}(\mathbf{x}, \mathbf{y})$$
$$\text{s.t.} \quad \mathbf{y} \text{ forms a tree}$$

We need to ensure that the colored edges form a valid output (i.e. a tree)

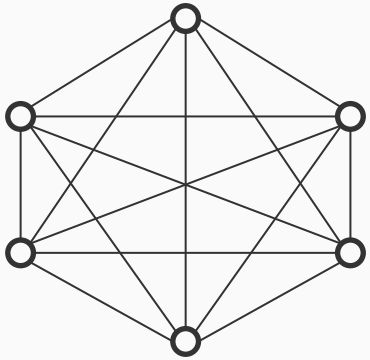




Combining local classifiers

Inference can "glue" together local decisions

– And enforce *global* coherence

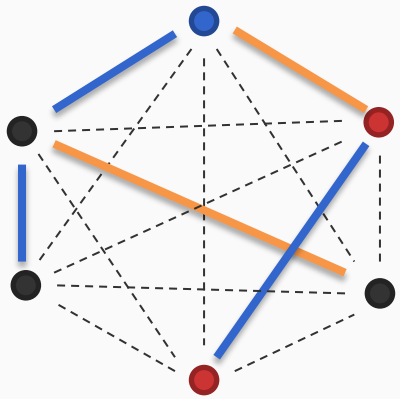


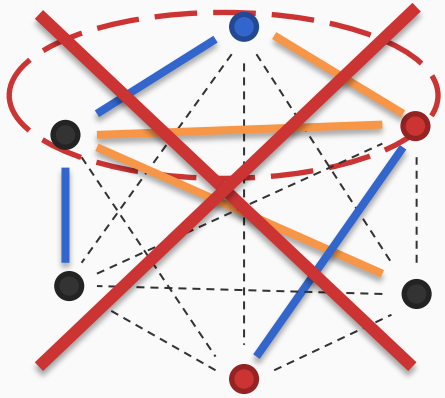
3 possible node labels

3 possible edge labels

$$\arg\max_{\mathbf{y}} \quad \text{score}(\mathbf{x}, \mathbf{y})$$
$$\text{s.t.} \quad \mathbf{y} \text{ forms a tree}$$

We need to ensure that the colored edges form a valid output (i.e. a tree)





Setting

Output: Nodes and edges are labeled and the blue and orange edges form a tree

Goal: Find the highest scoring tree

Constraints at prediction time

Introduce additional information

- Might not have been available at training time

Add domain knowledge

Examples:

- "All part-of-speech tag sequences should contain a verb"
- "Every bicycle should have at least one wheel"
- "In any window of size six, at least one of the labels should be a B"

Enforce coherence into the set of local decisions

Examples:

- "the collection of decisions should form a tree"
- "the collection of parts recognized should form a valid bicycle"

Outline

- Consistency of outputs and the value of inference
- Constrained conditional models via an example
- Hard constraints and Integer Programs
- Soft constraints

Constrained Conditional Model

Inference consists of two components

1. Local classifiers
   - Important: These may be a collection of structures themselves
   - These are trained models

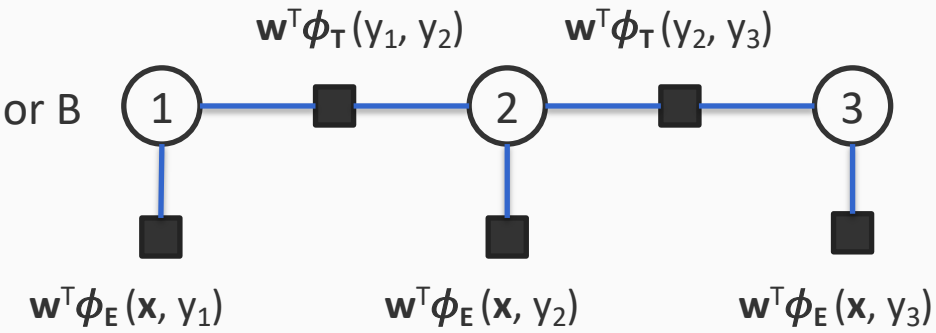
2. A set of constraints that restrict the space of joint assignments of the local classifiers
   - Background knowledge

An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B

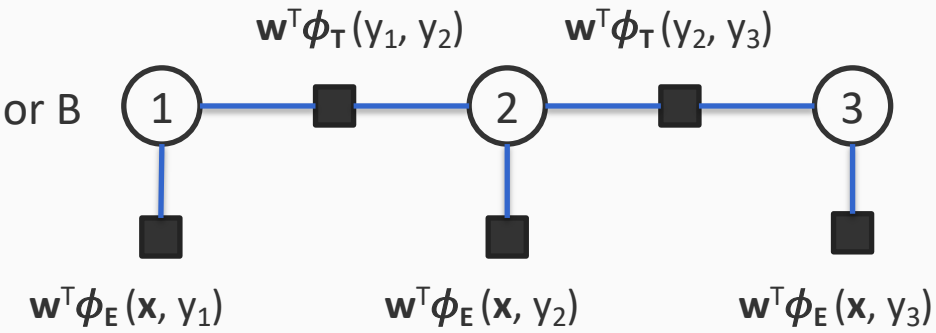
Typically, we have $\underset{\mathbf{y}}{\operatorname{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$



An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\operatorname{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

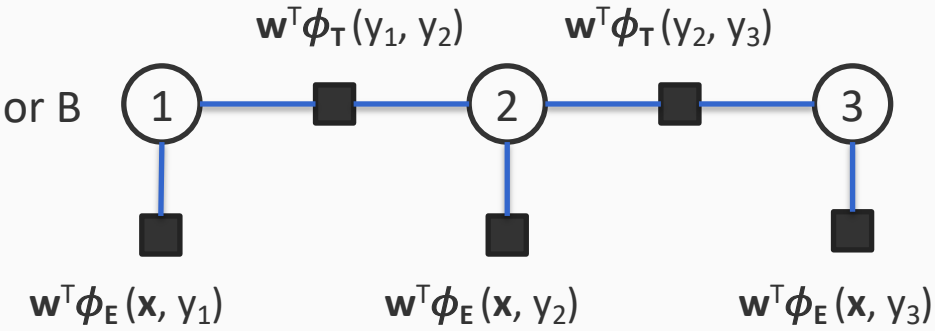
Or equivalently,

$$\underset{\mathbf{y}}{\operatorname{argmax}} \sum_i \mathbf{w}^T \phi_i(\mathbf{x}, \mathbf{y}_i)$$

An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\operatorname{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Or equivalently,

$$\underset{\mathbf{y}}{\operatorname{argmax}} \sum_i \mathbf{w}^T \phi_i(\mathbf{x}, \mathbf{y}_i)$$

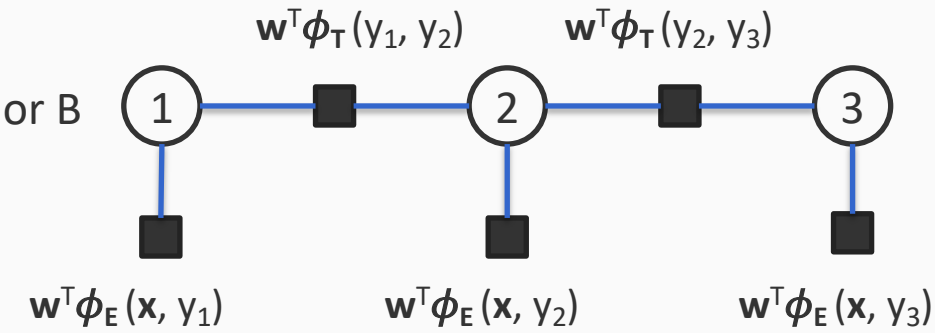
Sum over all parts/factors

Score for each factor

An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_i \mathbf{w}^T \phi_i(\mathbf{x}, \mathbf{y}_i)$$

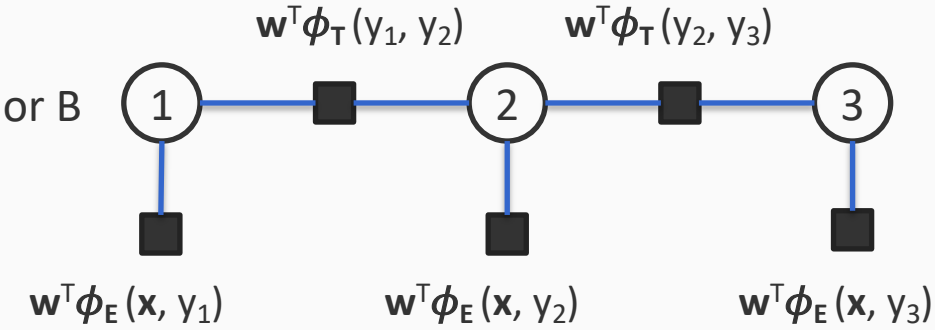
Or equivalently (here),

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \mathbf{w}^T \phi_E(\mathbf{x}, \mathbf{y}_i) + \sum_{i=1}^{2} \mathbf{w}^T \phi(\mathbf{y}_i, \mathbf{y}_{i+1})$$

An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_i \mathbf{w}^T \phi_i(\mathbf{x}, \mathbf{y}_i)$$

Or equivalently (here),

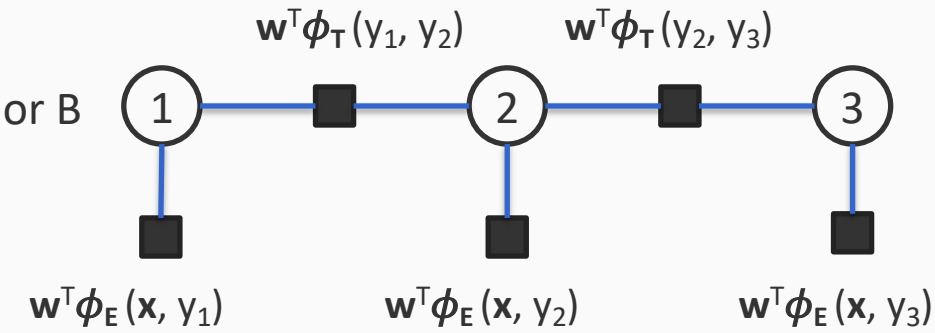
$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \mathbf{w}^T \phi_E(\mathbf{x}, \mathbf{y}_i) + \sum_{i=1}^{2} \mathbf{w}^T \phi(\mathbf{y}_i, \mathbf{y}_{i+1})$$

Emissions Transitions

An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \mathbf{w}^T \phi_E(\mathbf{x}, \mathbf{y}_i) + \sum_{i=1}^{2} \mathbf{w}^T \phi(\mathbf{y}_i, \mathbf{y}_{i+1})$$

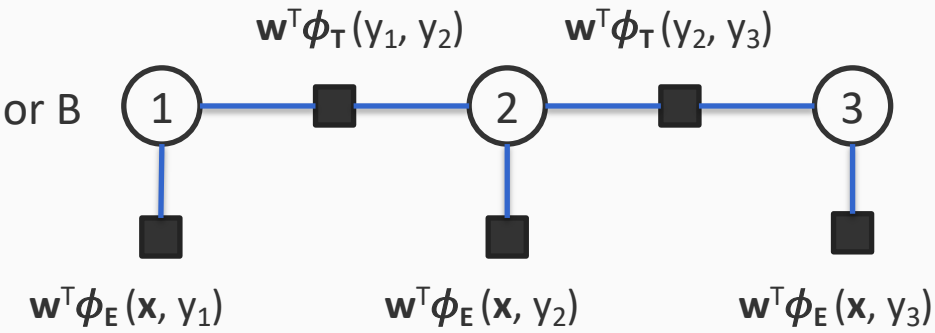
Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \mathbf{w}^T \phi_E(\mathbf{x}, \mathbf{y}_i) + \sum_{i=1}^{2} \mathbf{w}^T \phi(\mathbf{y}_i, \mathbf{y}_{i+1})$$

Or equivalently,

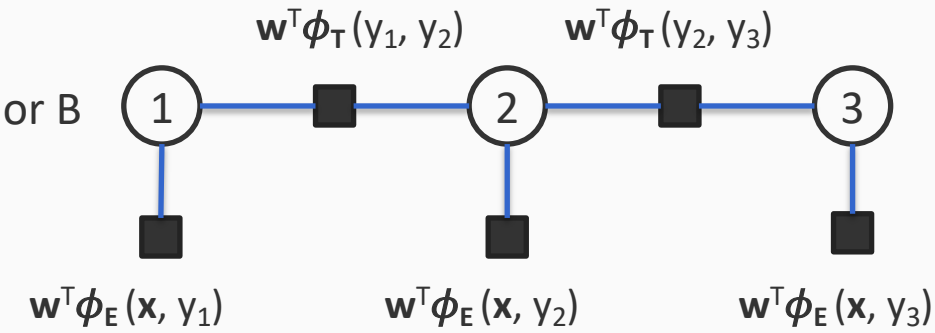
$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

Emissions

An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \mathbf{w}^T \phi_E(\mathbf{x}, \mathbf{y}_i) + \sum_{i=1}^{2} \mathbf{w}^T \phi(\mathbf{y}_i, \mathbf{y}_{i+1})$$

Or equivalently,

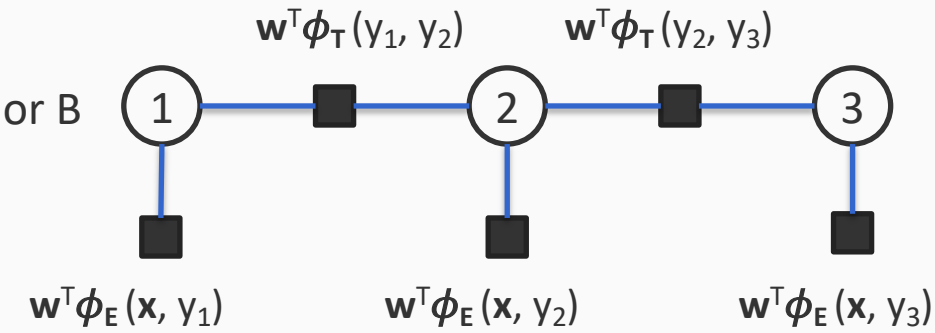
$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

Transitions

An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\operatorname{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Or equivalently,

$$\underset{\mathbf{y}}{\operatorname{argmax}} \sum_{i=1}^{3} \mathbf{w}^T \phi_E(\mathbf{x}, \mathbf{y}_i) + \sum_{i=1}^{2} \mathbf{w}^T \phi(\mathbf{y}_i, \mathbf{y}_{i+1})$$

Or equivalently,

$$\underset{\mathbf{y}}{\operatorname{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

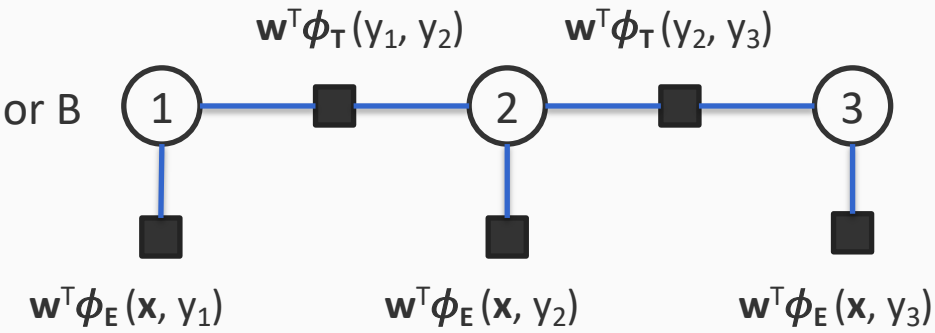
Indicators

$$I_z = \begin{cases} 1, & \text{if } z \text{ is true,} \\ 0, & \text{else.} \end{cases}$$

An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \mathbf{w}^T \phi_E(\mathbf{x}, \mathbf{y}_i) + \sum_{i=1}^{2} \mathbf{w}^T \phi(\mathbf{y}_i, \mathbf{y}_{i+1})$$

Or equivalently,

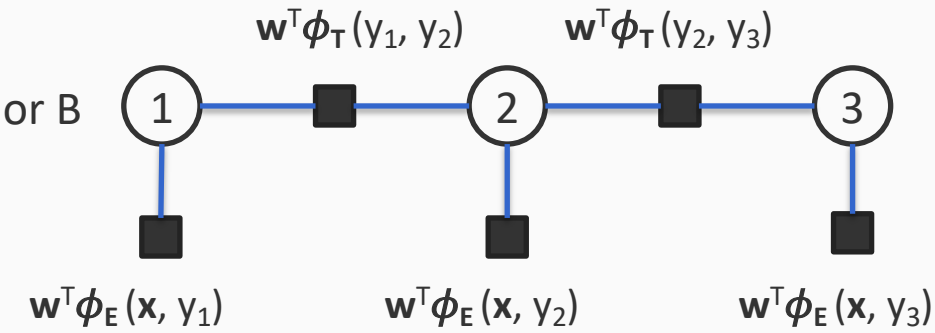
$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

Indicator for whether the i^{th} label is l.

An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \mathbf{w}^T \phi_E(\mathbf{x}, \mathbf{y}_i) + \sum_{i=1}^{2} \mathbf{w}^T \phi(\mathbf{y}_i, \mathbf{y}_{i+1})$$

Or equivalently,

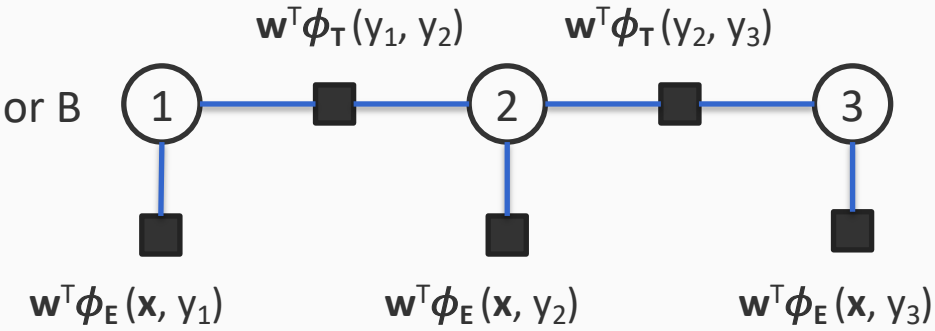
$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

Score for this decision,
typically from a trained model

An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \mathbf{w}^T \phi_E(\mathbf{x}, \mathbf{y}_i) + \sum_{i=1}^{2} \mathbf{w}^T \phi(\mathbf{y}_i, \mathbf{y}_{i+1})$$

Or equivalently,

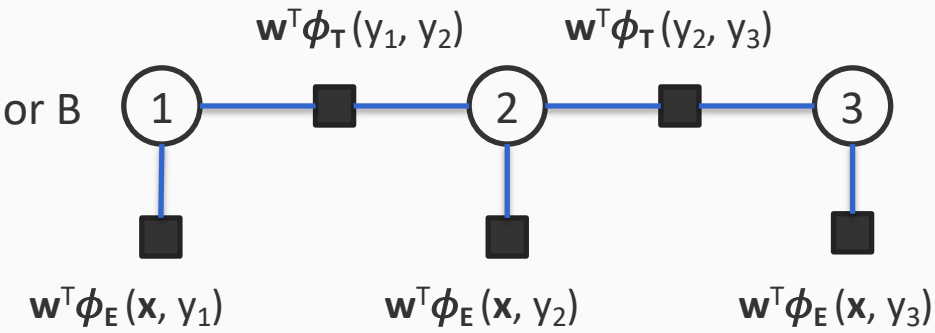
$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2} \mathbf{w}^T \phi(l_1, l_2)$$

Indicator for whether the i^{th} label is l_1 and the $(i+1)^{\text{th}}$ label is l_2.

An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \mathbf{w}^T \phi_E(\mathbf{x}, \mathbf{y}_i) + \sum_{i=1}^{2} \mathbf{w}^T \phi(\mathbf{y}_i, \mathbf{y}_{i+1})$$

Or equivalently,

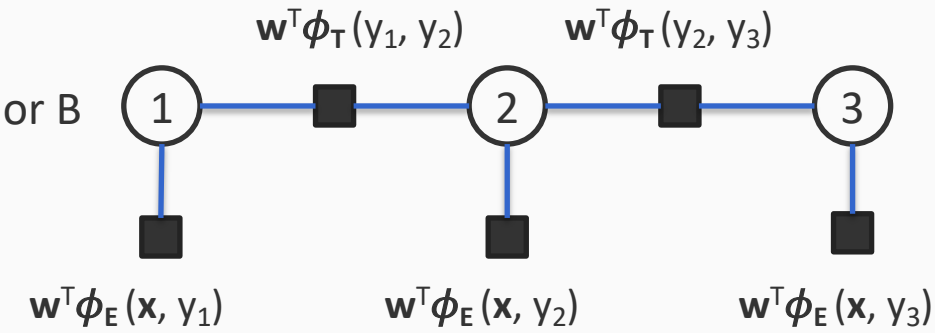
$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

Score for this decision, typically from a trained model

An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

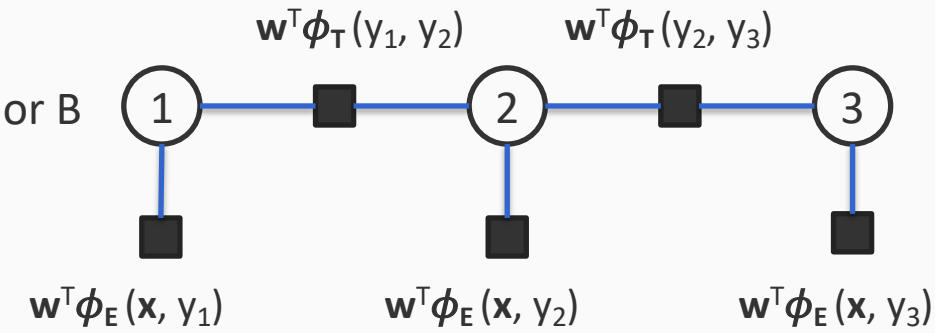
One group of indicators per factor

One score per indicator

An example of how we can formalize prediction

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

One group of indicators per factor

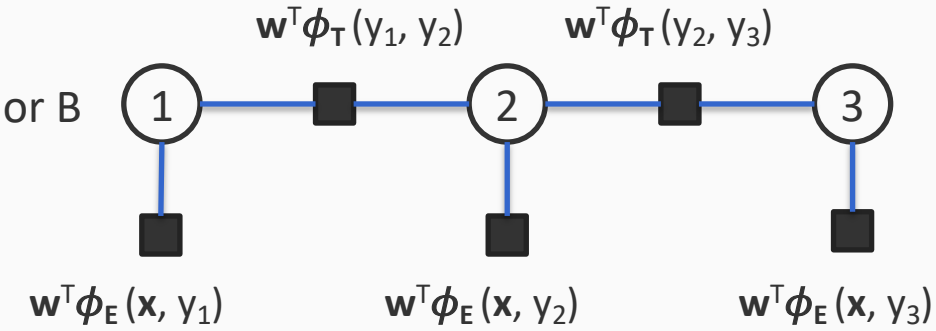
One score per indicator

This expression explicitly enumerates ***every*** decision that we need to make to build the final output

Prediction in a CCM

Suppose the outputs can be one of A or B

Typically, we have $\underset{\mathbf{y}}{\operatorname{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

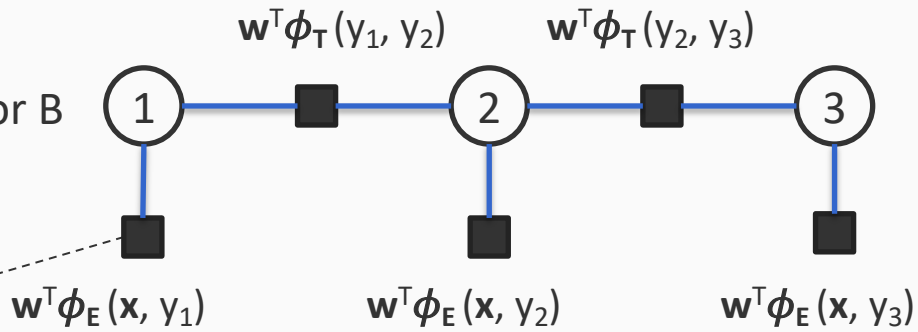


Let us enumerate ***every*** decision that we need to make to build the final output

Prediction in a CCM

Suppose the outputs can be one of A or B

Typically, we have $\underset{\mathbf{y}}{\operatorname{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$



Only one of these two decisions can exist

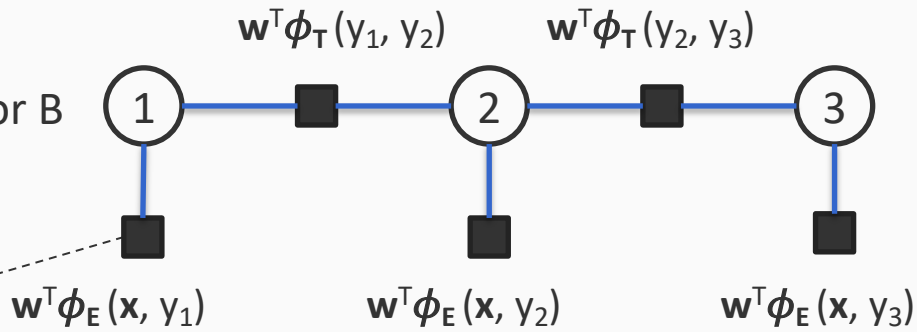
$I_{y_1=A}$ score(y_1=A) + $I_{y_1=B}$ score(y_1=B)

Let us enumerate ***every*** decision that we need to make to build the final output

Prediction in a CCM

Suppose the outputs can be one of A or B

Typically, we have $\underset{\mathbf{y}}{\operatorname{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$



Only one of these two decisions can exist

$I_{y_1=A}$ score(y_1=A) + $I_{y_1=B}$ score(y_1=B)

+ $I_{y_2=A}$ score(y_2=A) + $I_{y_2=B}$ score(y_2=B)

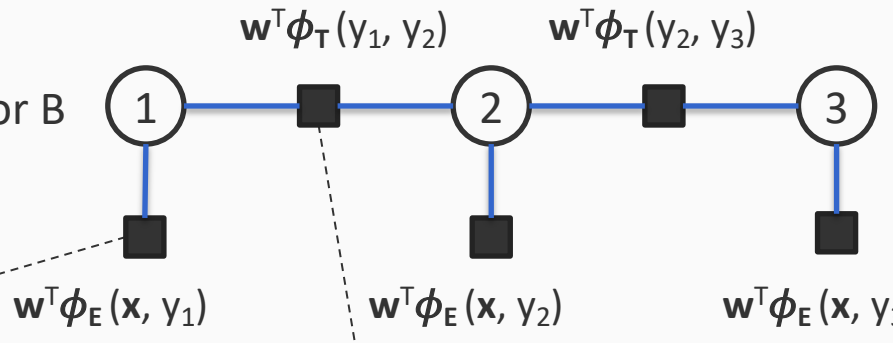
+ $I_{y_3=A}$ score(y_2=A) + $I_{y_3=B}$ score(y_2=B)

Let us enumerate ***every*** decision that we need to make to build the final output

Prediction in a CCM

Suppose the outputs can be one of A or B

Typically, we have $\underset{\mathbf{y}}{\operatorname{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$



Only one of these two decisions can exist

Only one of these four decisions can exist

$I_{y_{1=A}}$ score(y_1=A) + $I_{y_{1=B}}$ score(y_1=B)

+ $I_{y_{2=A}}$ score(y_2=A) + $I_{y_{2=B}}$ score(y_2=B)

+ $I_{y_{3=A}}$ score(y_2=A) + $I_{y_{3=B}}$ score(y_2=B)

+ $I_{y_{1=A \text{ and } y2=A}}$ score(y_1=A, y_2=A) + $I_{y_{1=A \text{ and } y2=B}}$ score(y_1=A, y_2=B)

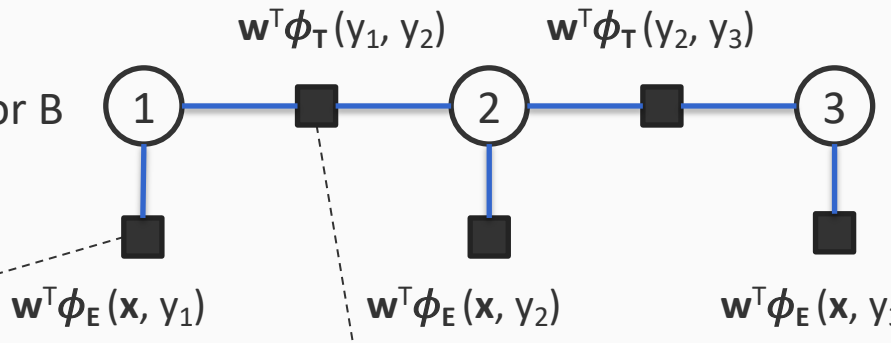
+ $I_{y_{1=B \text{ and } y2=A}}$ score(y_1=B, y_2=A) + $I_{y_{1=B \text{ and } y2=B}}$ score(y_1=B, y_2=B)

Let us enumerate ***every*** decision that we need to make to build the final output

Prediction in a CCM

Suppose the outputs can be one of A or B

Typically, we have $\underset{\mathbf{y}}{\operatorname{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$



Only one of these two decisions can exist

Only one of these four decisions can exist

$l_{y_1=A}$ score(y_1=A) + $l_{y_1=B}$ score(y_1=B)

+ $l_{y_2=A}$ score(y_2=A) + $l_{y_2=B}$ score(y_2=B)

+ $l_{y_3=A}$ score(y_2=A) + $l_{y_3=B}$ score(y_2=B)

+ $l_{y_1=A \text{ and } y2=A}$ score(y_1=A, y_2=A) + $l_{y_1=A \text{ and } y2=B}$ score(y_1=A, y_2=B)

+ $l_{y_1=B \text{ and } y2=A}$ score(y_1=B, y_2=A) + $l_{y_1=B \text{ and } y2=B}$ score(y_1=B, y_2=B)

+ $l_{y_2=A \text{ and } y3=A}$ score(y_2=A, y_3=A) + $l_{y_2=A \text{ and } y3=B}$ score(y_2=A, y_3=B)

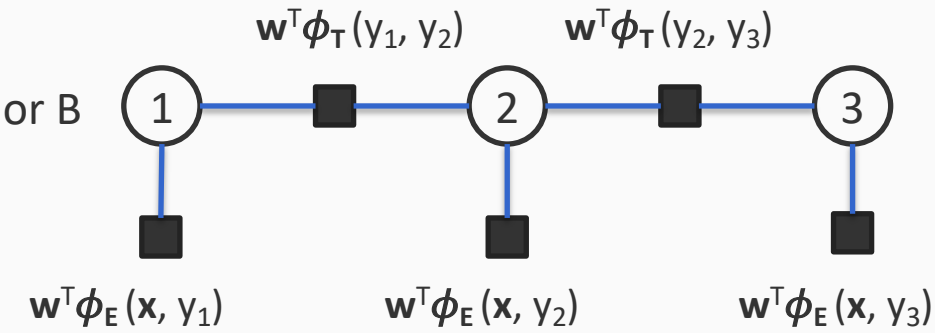
+ $l_{y_2=B \text{ and } y3=A}$ score(y_2=B, y_3=A) + $l_{y_2=B \text{ and } y3=B}$ score(y_2=B, y_3=B)

Let us enumerate ***every*** decision that we need to make to build the final output

Prediction in a CCM

Suppose the outputs can be one of A or B

Typically, we have $\underset{\mathbf{y}}{\operatorname{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$



$I_{y_{1=A}}$ score(y_1=A) + $I_{y_{1=B}}$ score(y_1=B)

+ $I_{y_{2=A}}$ score(y_2=A) + $I_{y_{2=B}}$ score(y_2=B)

+ $I_{y_{3=A}}$ score(y_2=A) + $I_{y_{3=B}}$ score(y_2=B)

+ $I_{y_{1=A \text{ and } y2=A}}$ score(y_1=A, y_2=A) + $I_{y_{1=A \text{ and } y2=B}}$ score(y_1=A, y_2=B)
+ $I_{y_{1=B \text{ and } y2=A}}$ score(y_1=B, y_2=A) + $I_{y_{1=B \text{ and } y2=B}}$ score(y_1=B, y_2=B

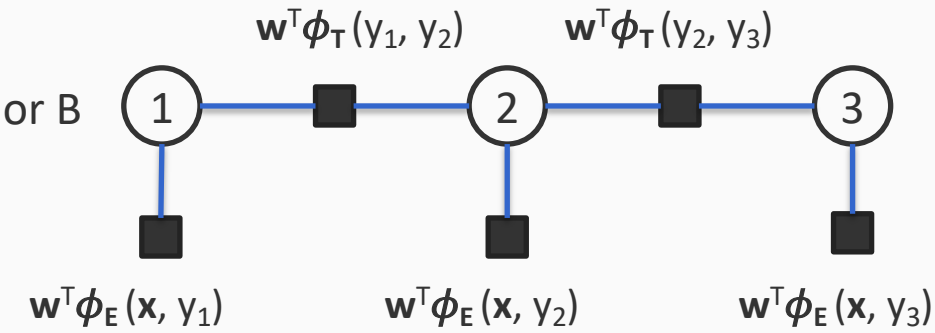
+ $I_{y_{2=A \text{ and } y3=A}}$ score(y_2=A, y_3=A) + $I_{y_{2=A \text{ and } y3=B}}$ score(y_2=A, y_3=B)
+ $I_{y_{2=B \text{ and } y3=A}}$ score(y_2=B, y_3=A) + $I_{y_{2=B \text{ and } y3=B}}$ score(y_2=B, y_3=B

Some decisions can not exist together

Prediction in a CCM

Suppose the outputs can be one of A or B

Typically, we have $\underset{\mathbf{y}}{\operatorname{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$



$I_{y_1=A}$ score(y_1=A) + $I_{y_1=B}$ score(y_1=B)

+ $I_{y_2=A}$ score(y_2=A) + $I_{y_2=B}$ score(y_2=B)

+ $I_{y_3=A}$ score(y_2=A) + $I_{y_3=B}$ score(y_2=B)

+ $I_{y_1=A \text{ and } y2=A}$ score(y_1=A, y_2=A) + $I_{y_1=A \text{ and } y2=B}$ score(y_1=A, y_2=B)

+ $I_{y_1=B \text{ and } y2=A}$ score(y_1=B, y_2=A) + $I_{y_1=B \text{ and } y2=B}$ score(y_1=B, y_2=B

+ $I_{y_2=A \text{ and } y3=A}$ score(y_2=A, y_3=A) + $I_{y_2=A \text{ and } y3=B}$ score(y_2=A, y_3=B)

+ $I_{y_2=B \text{ and } y3=A}$ score(y_2=B, y_3=A) + $I_{y_2=B \text{ and } y3=B}$ score(y_2=B, y_3=B

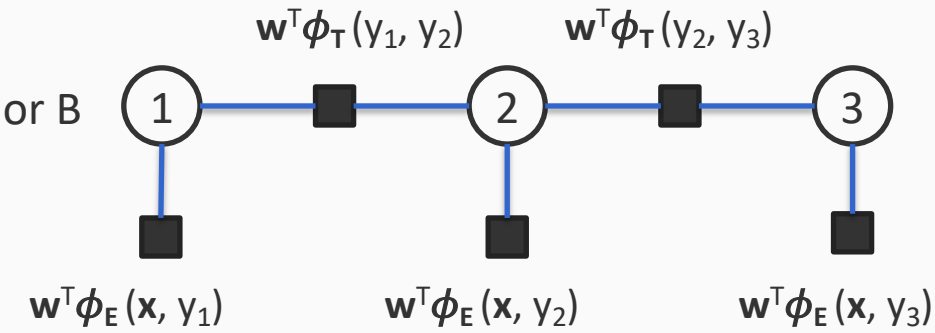
Some decisions can not exist together

If $I_{y_2=A} = 1$ then $I_{y_2=B \text{ and } y_3=A} = 0$

If $I_{y_2=A} = 1$ then $I_{y_1=B \text{ and } y_2=B} = 0$

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

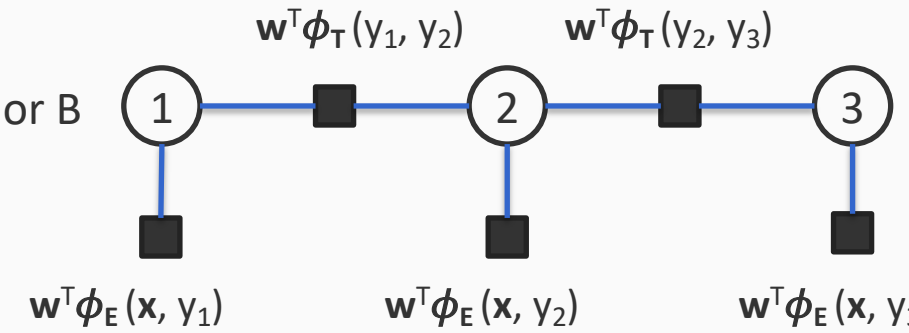
Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

But we should only consider valid label sequences

Prediction in a CCM

Suppose the outputs can be one of A or B



Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$

Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

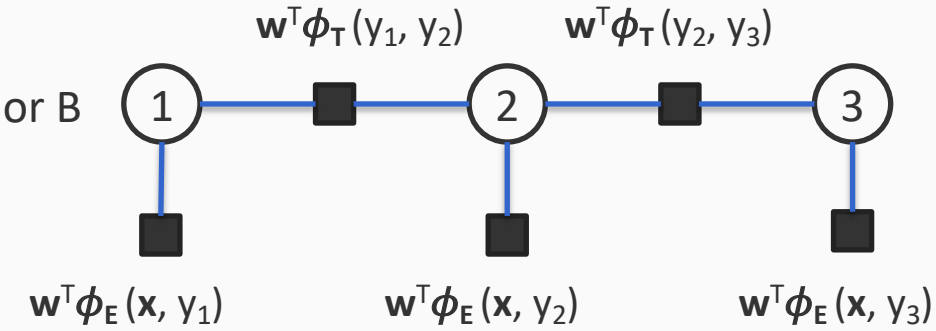
Predict with constraints

- Each y_i can either be a A or a B

Prediction in a CCM

Suppose the outputs can be one of A or B

Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$



Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

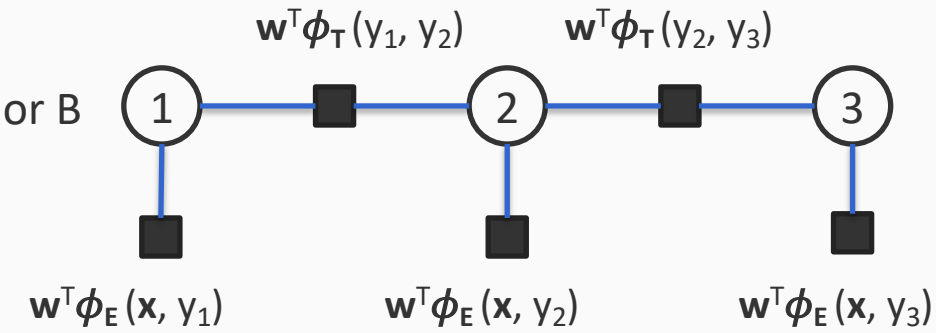
Predict with constraints

- Each y_i can either be a A or a B: $\forall i, I_{y_i=A} + I_{y_i=B} = 1$

Prediction in a CCM

Suppose the outputs can be one of A or B

Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$



Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

Predict with constraints

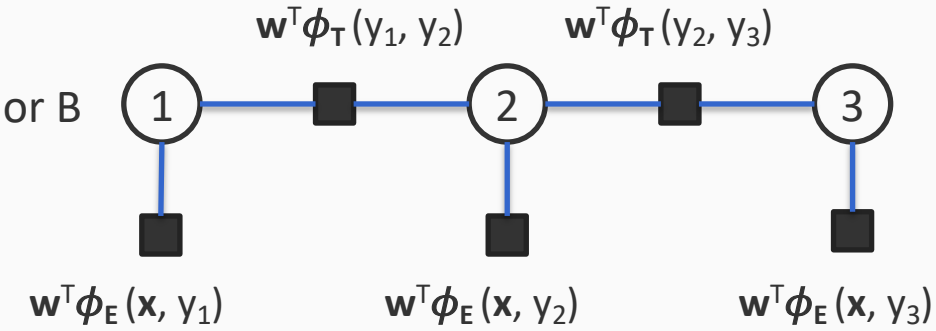
- Each y_i can either be a A or a B: $\forall i, I_{y_i=A} + I_{y_i=B} = 1$
- The emission decisions and the transition decisions should agree

We can write this using linear constraints over the indicator variables

Prediction in a CCM

Suppose the outputs can be one of A or B

Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$



Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

Predict with constraints

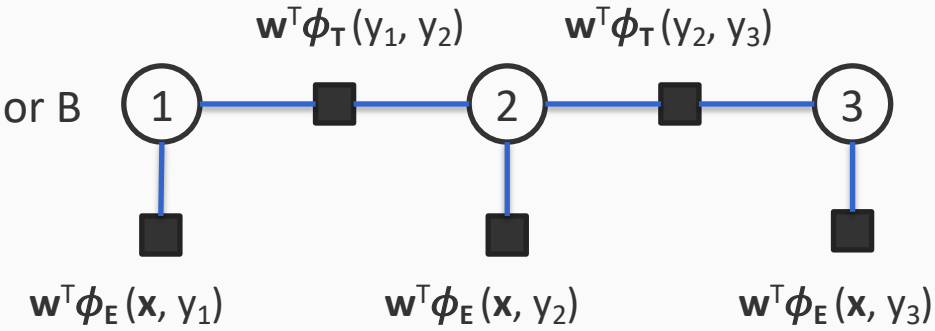
- Each y_i can either be a A or a B: $\forall i, I_{y_i=A} + I_{y_i=B} = 1$
- The emission decisions and the transition decisions should agree
- There should be no more than one B in the output

We could add extra knowledge that was not present at training time

Prediction in a CCM

Suppose the outputs can be one of A or B

Typically, we have $\underset{\mathbf{y}}{\text{argmax}}\, \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y})$



Or equivalently,

$$\underset{\mathbf{y}}{\text{argmax}} \sum_{i=1}^{3} \sum_{l \in \{A,B\}} I_{y_i = l} \cdot \mathbf{w}^T \phi_E(\mathbf{x}, l) + \sum_{i=1}^{2} \sum_{l_1, l_2 \in \{A,B\}} I_{y_i = l_1 \wedge y_{i+1} = l_2}\, \mathbf{w}^T \phi(l_1, l_2)$$

Predict with constraints

- Each y_i can either be a A or a B: $\forall i, I_{y_i=A} + I_{y_i=B} = 1$
- The emission decisions and the transition decisions should agree
- There should be no more than one B in the output

$$I_{y_1=B} + I_{y_2=B} + I_{y_3=B} \leq 1$$

Questions?

Constrained Conditional Model

with hard constraints

- Global joint inference
  - Treat the output as a collection of decisions (one per factor/part)
  - Each decision associated with a score
  - In addition, allow arbitrary constraints involving the decisions

- Constraints
  - Inject domain knowledge into the prediction
  - Can be stated as logical statements
  - Can be transformed into linear inequalities in terms of the decision variables

- No comment about learning
  - Learn the scoring functions locally or globally

Outline

- Consistency of outputs and the value of inference
- Constrained conditional models via an example
- Hard constraints and Integer Programs
- Soft constraints

Inference with hard constraints

$$\arg\max_{\mathbf{y}} \sum_{p \in \text{Parts}(\mathbf{x})} \sum_{l \in \text{Labels}} I_{y_p = l} \cdot \text{score}(\mathbf{x}, l)$$

Such that **y** is feasible

Inference with hard constraints

$$\arg\max_{\mathbf{y}} \sum_{p \in \text{Parts}(\mathbf{x})} \sum_{l \in \text{Labels}} I_{y_p = l} \cdot \text{score}(\mathbf{x}, l)$$

Such that **y** is feasible

Same as

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

Inference with hard constraints

$$\arg\max_{\mathbf{y}} \sum_{p \in \text{Parts}(\mathbf{x})} \sum_{l \in \text{Labels}} I_{y_p = l} \cdot \text{score}(\mathbf{x}, l)$$

Such that **y** is feasible

Same as

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

- Can be written as linear (in)equalities in the I's
- I's can only be 0 or 1

This is an Integer Linear Program

Inference with hard constraints

$$\arg\max_{\mathbf{y}} \sum_{p \in \text{Parts}(\mathbf{x})} \sum_{l \in \text{Labels}} I_{y_p = l} \cdot \text{score}(\mathbf{x}, l)$$

Such that **y** is feasible

Same as

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

- Can be written as linear (in)equalities in the I's
- I's can only be 0 or 1

This is an Integer Linear Program

MAP inference can be written as integer linear programs (ILPs)

Inference with hard constraints

$$\arg\max_{\mathbf{y}} \sum_{p \in \text{Parts}(\mathbf{x})} \sum_{l \in \text{Labels}} I_{y_p = l} \cdot \text{score}(\mathbf{x}, l)$$

Such that **y** is feasible

Same as

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

- Can be written as linear (in)equalities in the I's
- I's can only be 0 or 1

This is an Integer Linear Program

MAP inference can be written as integer linear programs (ILPs)

But solving ILPs is NP-complete in the worst case

- Use an off-the-shelf solver (Gurobi) and hope for the best
- **Or not**: Write a specialized search algorithm if we know more about the problem (exact or approximate)
- *We will see examples of these*

Questions?

ILP inference

- Can introduce domain knowledge in the form of constraints
  - Any Boolean expression over the inference variables can be written as linear inequalities
- A uniform language for formulating and reasoning about inference
- Have not made the problem any easier to solve
  - By allowing easy addition of constraints, it may be simple to write down provably intractable inference formulations
  - (Off-the-shelf solvers seem to work admirably!)

Questions?

Outline

- Consistency of outputs and the value of inference
- Constrained conditional models via an example
- Hard constraints and Integer Programs
- Soft constraints

Constraints may not always hold

"Every car should have a wheel"

Sometimes constraints don't always hold
Allow the model to consider structures that violate constraints

Constrained Conditional Model

General case: with soft constraints

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

The model score for the structure

Sometimes constraints don't always hold
Allow the model to consider structures that violate constraints

Constrained Conditional Model

General case: with soft constraints

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

The model score for the structure

Suppose we have a collection of "soft constraints" $C_1, C_2, \cdots$ each associated with penalties for violation $\rho_1, \rho_2, \cdots$

That is: A constraint C_k is a Boolean expression over the output. An assignment that violates this constraint has to pay a penalty of ρ_k

Sometimes constraints don't always hold
Allow the model to consider structures that violate constraints

Constrained Conditional Model

General case: with soft constraints

$$\arg\max_{\mathbf{y}} \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y}) - \sum_k \rho_k d(\mathbf{y}, I_{C_k(\mathbf{x})})$$

The model score for the structure

Penalty for violating the constraint

If the constraint is hard, penalty = 1.

Sometimes constraints don't always hold
Allow the model to consider structures that violate constraints

Constrained Conditional Model

General case: with soft constraints

$$\arg\max_{\mathbf{y}} \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y}) - \sum_k \rho_k d(\mathbf{y}, I_{C_k(\mathbf{x})})$$

The model score for the structure

Penalty for violating the constraint

If the constraint is hard, penalty = 1.

How far is the structure **y** from satisfying the constraint C_k

The most simple case:
0: If satisfied
1: If not

Sometimes constraints don't always hold
Allow the model to consider structures that violate constraints

Constrained conditional models: review

- Write down conditions that the output need to satisfy
  - Constraints are effectively factors in a factor graph whose potential functions are fixed
- Different learning regimes
  - Train with the constraints or without
  - Remember: constraint penalties are fixed in either case
- Prediction
  - Can write the inference formulation as an integer linear program
  - Can solve it with an off-the-shelf solver (or not!)
- Extension
  - Soft constraints*:* Constraints that don't always need to hold

Questions?

Structured prediction: General formulations

- Graphical models
  - Bayesian Networks
  - Markov Random Fields
- Formulations of structured output
  - Joint models
    - Markov Logic Network
  - Conditional models
    - Conditional Random Fields (again)
    - Constrained Conditional Models

All the discussion so far has been about representation.

Forthcoming lectures: Algorithms for training and inference