Inference

CS 6355: Structured Prediction

THE UNIVERSITY OF UTAH

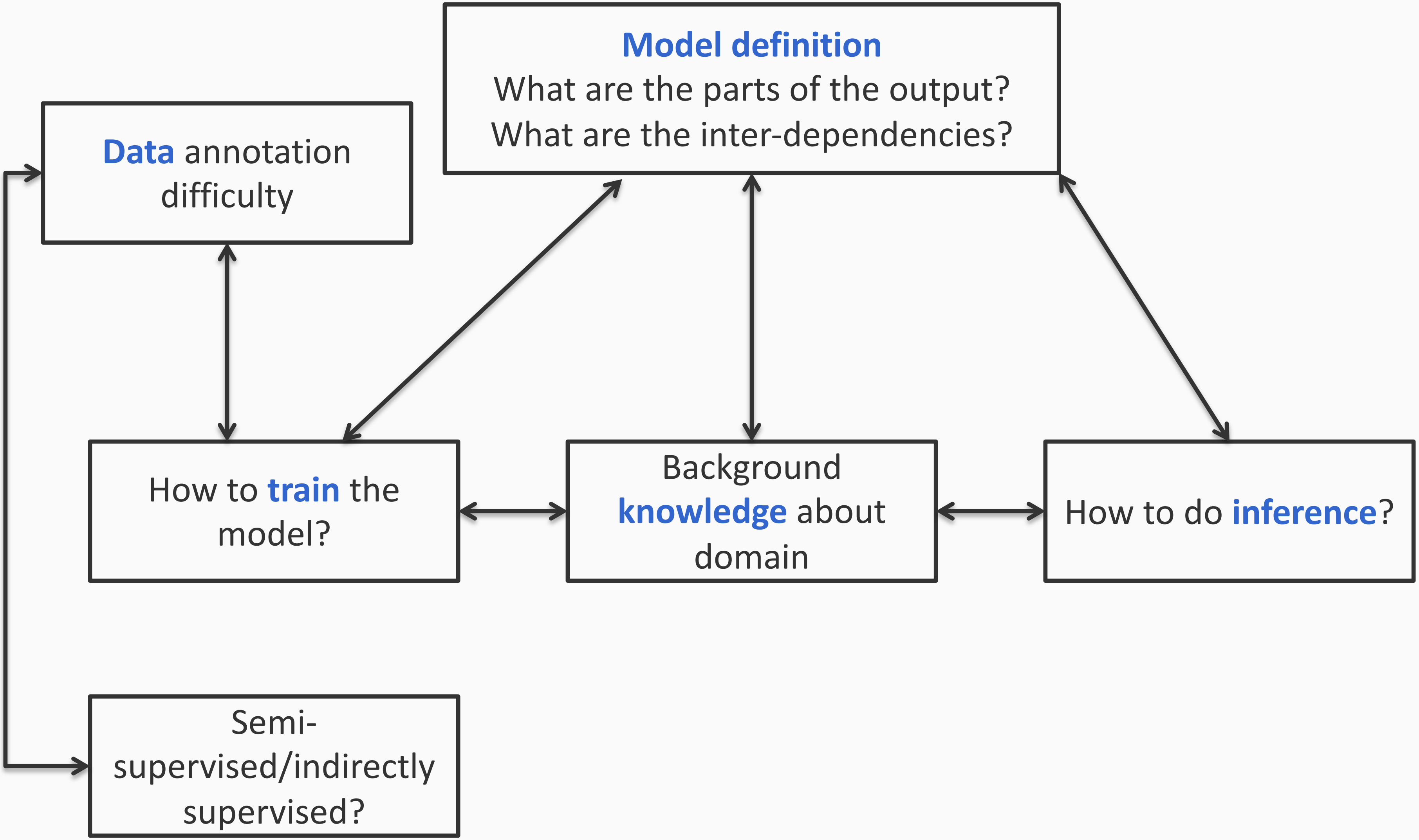
So far in the class

- Thinking about structures
  - A graph, a collection of parts that are labeled jointly, a collection of decisions
  - Key idea: Instead of assigning scores/probabilities to entire structures, construct the score by accumulating scores over parts.

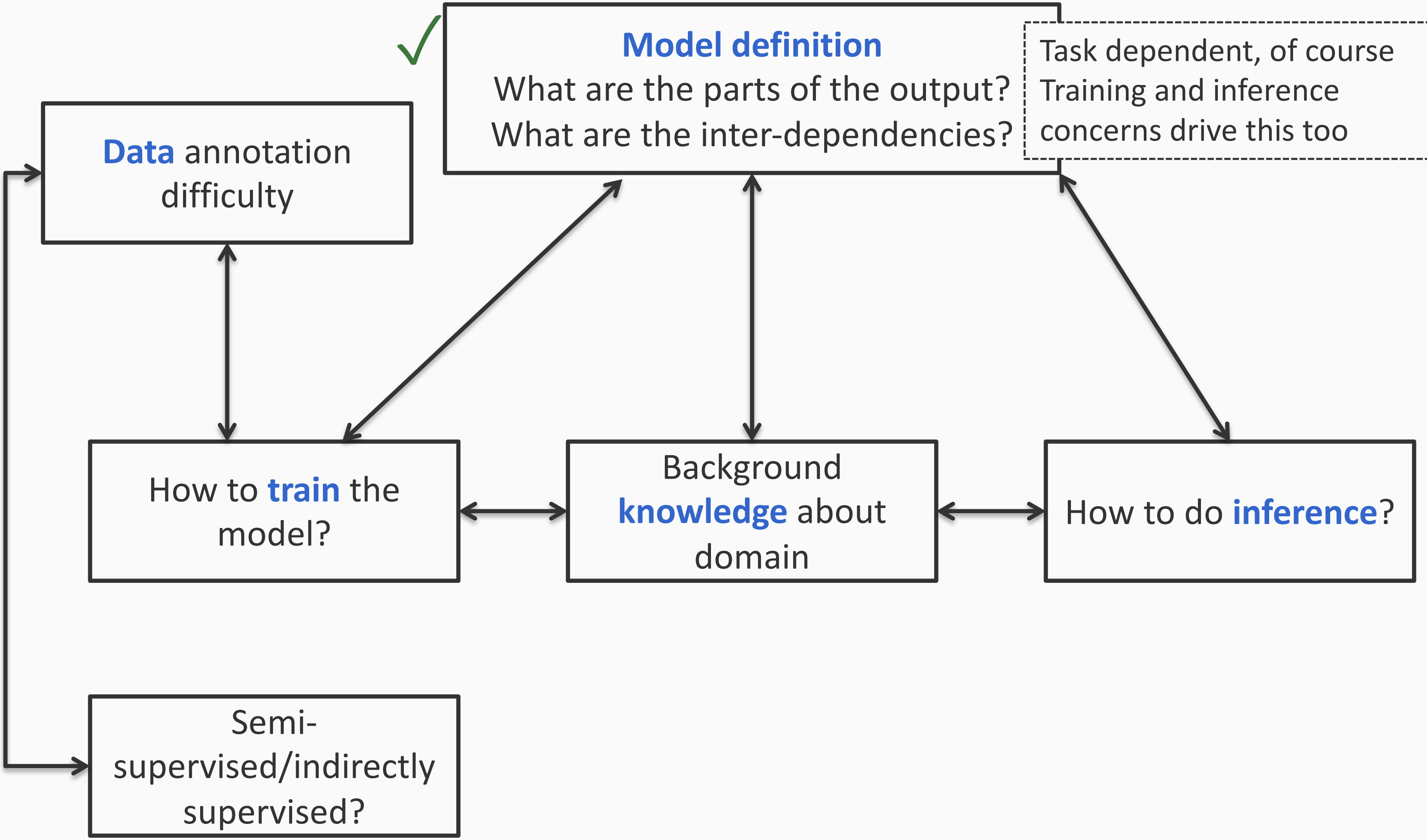
- Algorithms for learning
  - Local learning
    - Learn parameters for individual components independently
    - Learning algorithm not aware of the full structure
  - Global learning
    - Learn parameters for the full structure
    - Learning algorithm “knows” about the full structure

- Next: ***Prediction***
  - Sets structured prediction apart from binary/multiclass

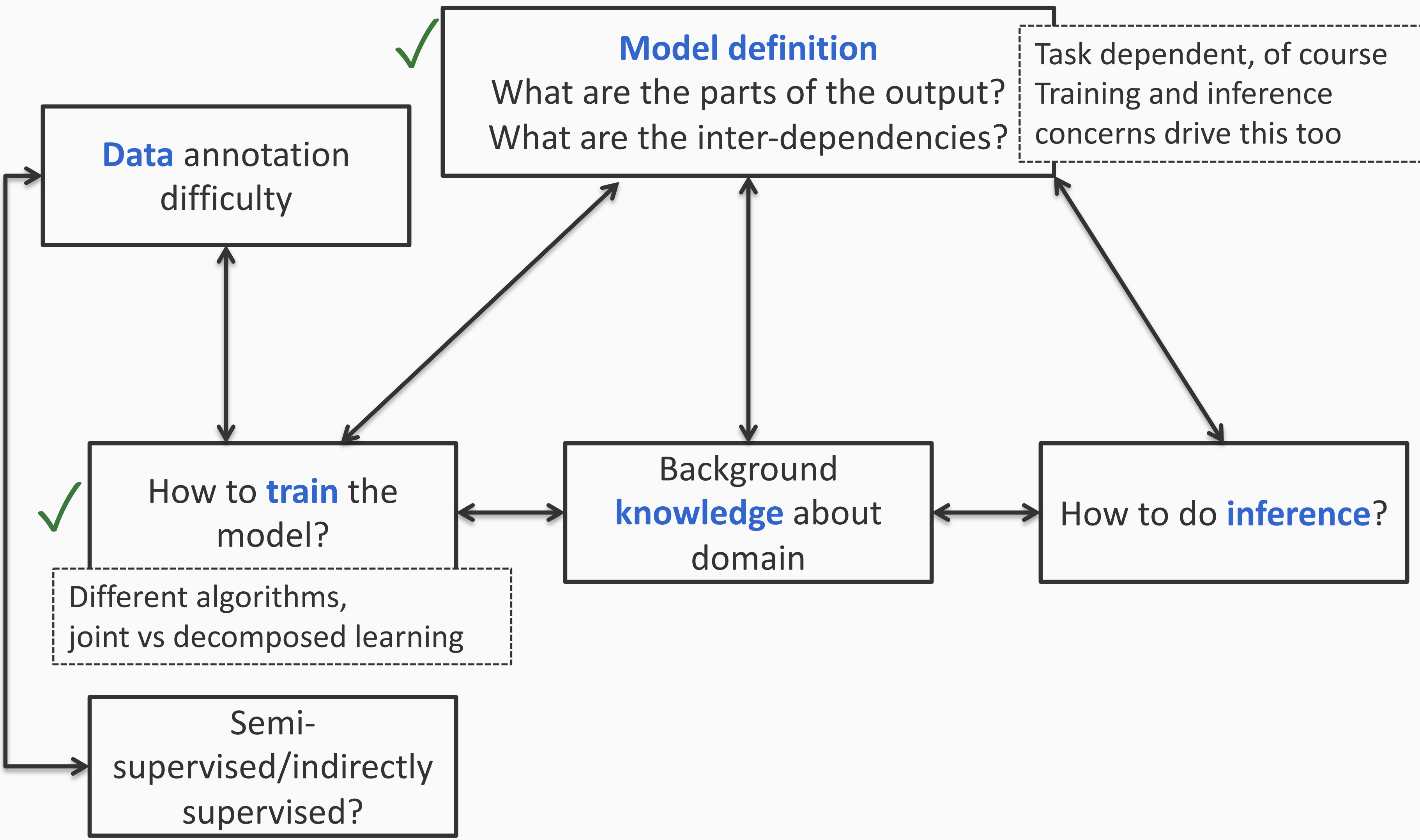
Computational issues



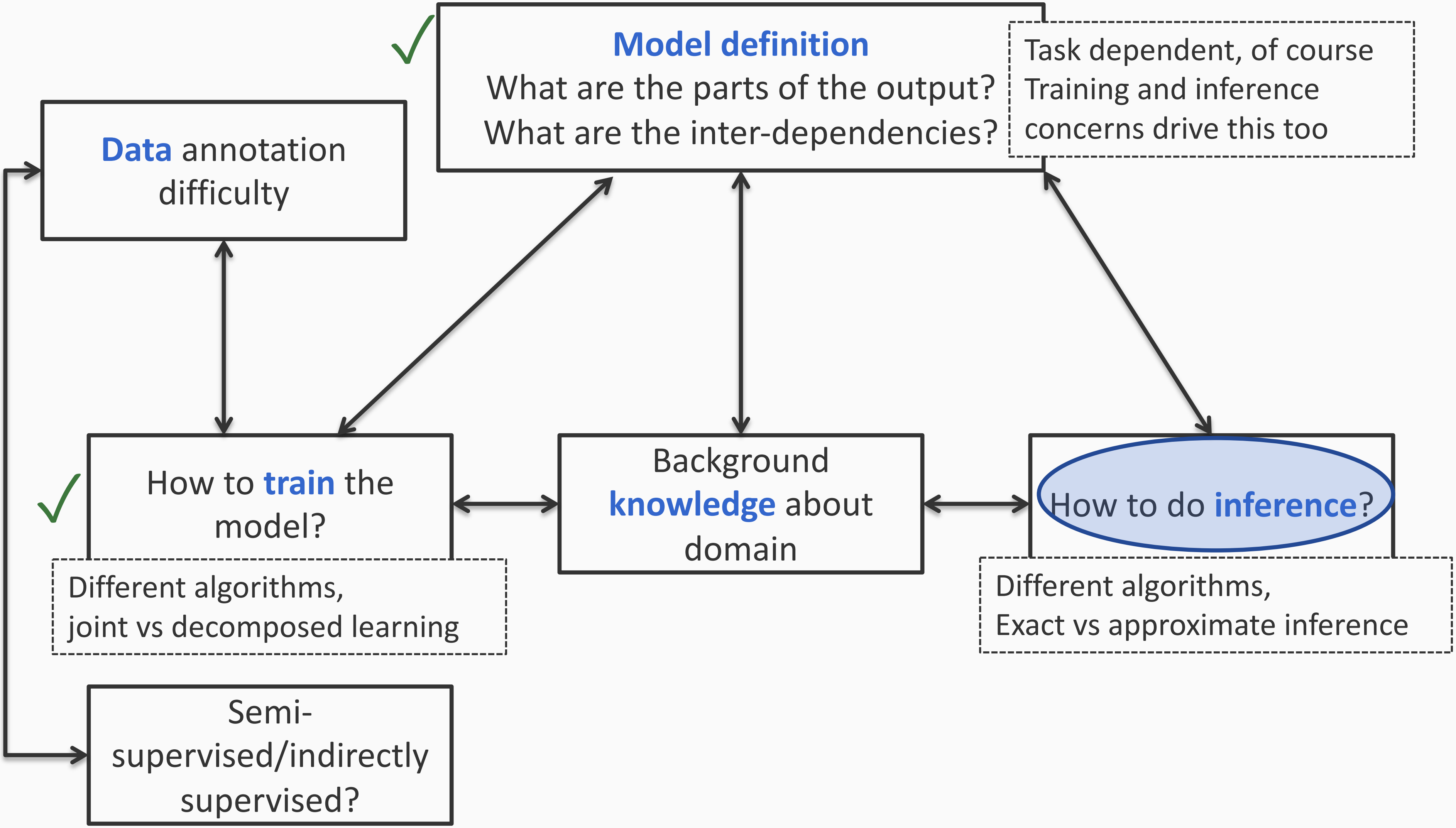
Computational issues



Computational issues



Computational issues



Inference

- What is inference?
  - An overview of what we have seen before
  - Combinatorial optimization
  - Different views of inference
- Graph algorithms
  - Dynamic programming, greedy algorithms, search
- Integer programming
- Heuristics for inference
  - Sampling
- Learning to search

Inference

- What is inference?
  - An overview of what we have seen before
  - Combinatorial optimization
  - Different views of inference

- Graph algorithms
  - Dynamic programming, greedy algorithms, search

- Integer programming

- Heuristics for inference
  - Sampling

- Learning to search

Remember sequence prediction

- Goal: Find the most probable/highest scoring state sequence

$$\operatorname*{argmax}_{\mathbf{y}} score(\mathbf{y}) = \operatorname*{argmax}_{\mathbf{y}} w^T \phi(\mathrm{x}, \mathbf{y})$$

  – Maximum a posteriori inference

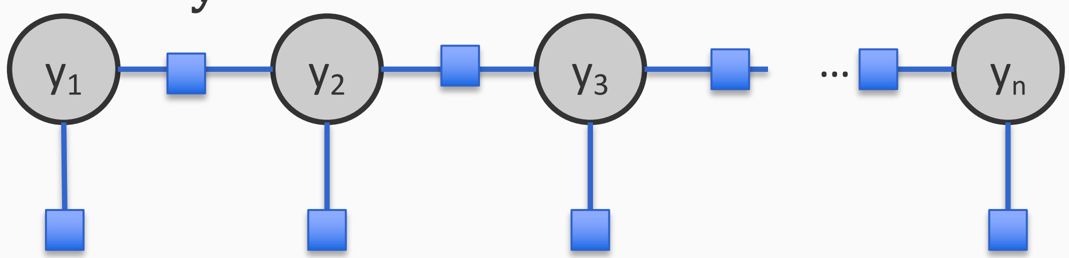
- Computationally: discrete optimization

- The naïve algorithm
  – Enumerate all sequences, score each one and pick the max
  – Terrible idea!

- We can do better
  – Scores decomposed over edges

The Viterbi algorithm: Recurrence

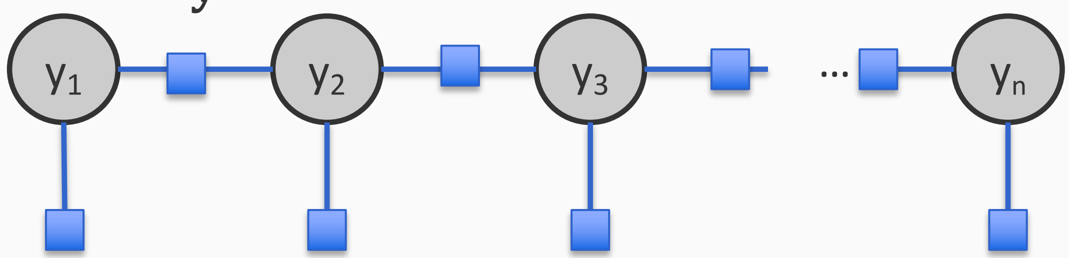
Goal: Find $\text{argmax}_{\mathbf{y}} \, w^T \phi(\mathrm{x}, \mathbf{y})$ $\mathbf{y} = (y_1, y_2, \cdots y_n)$



$$\mathbf{w}^T \phi(\mathbf{x}, \mathbf{y}) = \sum_i \left(\mathbf{w}^T \phi_T(y_i, y_{i+1}) + \mathbf{w}^T \phi_E(\mathbf{x}, y_i) \right)$$

The Viterbi algorithm: Recurrence

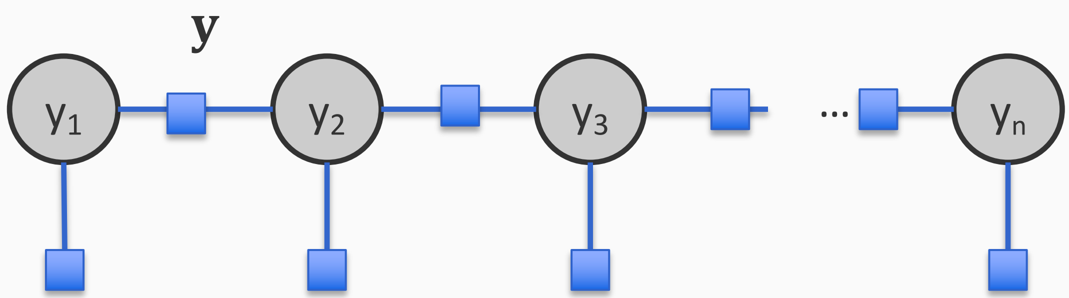
Goal: Find $\underset{\mathbf{y}}{\operatorname{argmax}}\, w^T \phi(\mathrm{x}, \mathbf{y})$ $\mathbf{y} = (y_1, y_2, \cdots y_n)$



$$\mathbf{w}^T \phi(\mathbf{x}, \mathbf{y}) = \sum_i \left(\mathbf{w}^T \phi_T(y_i, y_{i+1}) + \mathbf{w}^T \phi_E(\mathbf{x}, y_i)\right) = \sum_i \text{score-local}_i(y_i, y_{i+1})$$

The Viterbi algorithm: Recurrence

Goal: Find $\operatorname{argmax}_{\mathbf{y}} w^T \phi(x, \mathbf{y})$ $\mathbf{y} = (y_1, y_2, \cdots y_n)$



$$\mathbf{w}^T\phi(\mathbf{x},\mathbf{y}) = \sum_i \left(\mathbf{w}^T\phi_T(y_i, y_{i+1}) + \mathbf{w}^T\phi_E(\mathbf{x}, y_i)\right) = \sum_i \text{score-local}_i(y_i, y_{i+1})$$

$$\text{score}_1(s) = \text{score-local}_1(s, \text{START})$$
$$\text{score}_i(s) = \max_{y_{i-1}} \left[\text{score}_{i-1}(y_{i-1}) + \text{score-local}_i(s)\right]$$

Idea

1. If I know the score of all sequences y_1 to y_{n-1}, then I could decide y_n easily
2. Recurse to get score up to y_{n-1}

Building the answer

$$\text{score}_1(s) = \text{score-local}_1(s, \textbf{START})$$
$$\text{score}_i(s) = \max_{y_{i-1}} \left[\text{score}_{i-1}(y_{i-1}) + \text{score-local}_i(s)\right]$$

- $\text{score}_i(s)$ gets us the score for the best state sequence till the i^{th} position that ends in the state s
- What we want: The actual state sequence, not the score
- How do we construct it?

Building the answer

$$\text{score}_1(s) = \text{score-local}_1(s, \mathtt{START})$$
$$\text{score}_i(s) = \max_{y_{i-1}} \left[\text{score}_{i-1}(y_{i-1}) + \text{score-local}_i(s)\right]$$

- $\text{score}_i(s)$ gets us the score for the best state sequence till the i^{th} position that ends in the state s
- What we want: The actual state sequence, not the score
- How do we construct it?
  - Keep back pointers for the **max** at each step that tells you which state led us to the score at that step

Questions?

The bigger picture

- The goal of structured prediction: Predicting a graph
- Modeling: Defining probability distributions over the random variables
  - Involves making independence assumptions
- Learning creates functions that score predictions
- Inference: The computational step that actually constructs the output
  - Also called *decoding* in some papers

Inference questions

- This class:
  - Mostly we use inference to mean "*What is the highest scoring assignment to the output random variables for a given input?*"
  - Maximum A Posteriori (MAP) inference (if the score is probabilistic)

Inference questions

- This class:
  - Mostly we use inference to mean "*What is the highest scoring assignment to the output random variables for a given input?*"
  - Maximum A Posteriori (MAP) inference (if the score is probabilistic)

- Other inference questions exist

Inference questions

- This class:
  - Mostly we use inference to mean "*What is the highest scoring assignment to the output random variables for a given input?*"
  - Maximum A Posteriori (MAP) inference (if the score is probabilistic)

- Other inference questions exist
  - What is the highest scoring assignment to *some* of the output variables given the input?

Inference questions

- This class:
  - Mostly we use inference to mean "*What is the highest scoring assignment to the output random variables for a given input?*"
  - Maximum A Posteriori (MAP) inference (if the score is probabilistic)

- Other inference questions exist
  - What is the highest scoring assignment to *some* of the output variables given the input?
  - Sample from the posterior distribution over the **Y**

Inference questions

- This class:
  - Mostly we use inference to mean "*What is the highest scoring assignment to the output random variables for a given input?*"
  - Maximum A Posteriori (MAP) inference (if the score is probabilistic)

- Other inference questions exist
  - What is the highest scoring assignment to *some* of the output variables given the input?
  - Sample from the posterior distribution over the **Y**
  - Loss-augmented inference: Which structure most violates the margin for a given scoring function?

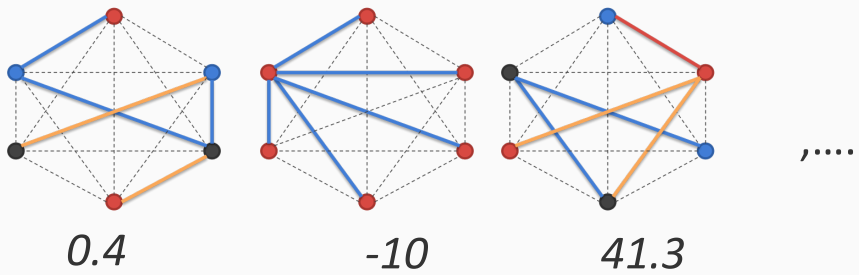
Inference questions

- This class:
  - Mostly we use inference to mean "*What is the highest scoring assignment to the output random variables for a given input?*"
  - Maximum A Posteriori (MAP) inference (if the score is probabilistic)

- Other inference questions exist
  - What is the highest scoring assignment to *some* of the output variables given the input?
  - Sample from the posterior distribution over the **Y**
  - Loss-augmented inference: Which structure most violates the margin for a given scoring function?
  - Computing marginal probabilities over **Y**

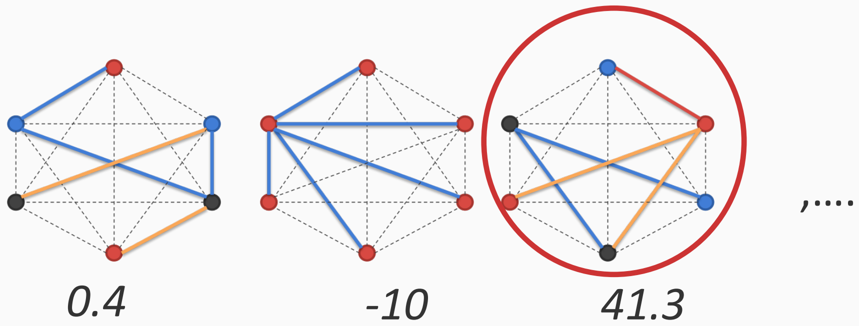
MAP inference

- A combinatorial problem



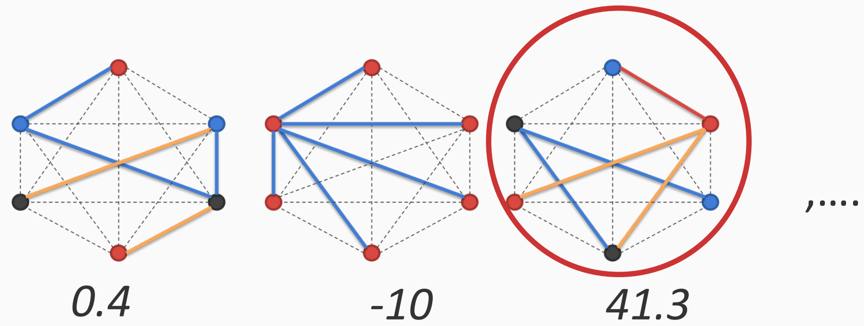
MAP inference

- A combinatorial problem



MAP inference is discrete optimization

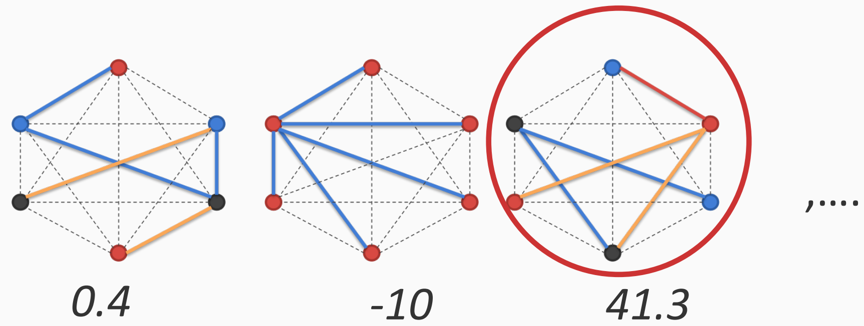
- A combinatorial problem



- Computational complexity depends on
  - The size of the input
  - The factorization of the scores
  - More complex factors generally lead to expensive inference

MAP inference is discrete optimization

- A combinatorial problem



- Computational complexity depends on
  - The size of the input
  - The factorization of the scores
  - More complex factors generally lead to expensive inference
- A generally bad strategy in most but the simplest cases:
  “Enumerate all possible structures and pick the highest scoring one”

MAP inference is search

- We want a collections of decisions that has highest score

$$\operatorname*{argmax}_{\mathbf{y}} w^T \phi(\mathrm{x}, \mathbf{y}) \qquad \mathbf{y} = (y_1, y_2, \cdots y_n)$$

- No algorithm can find the max without considering every possible structure
  - Why?

- Key question to consider: How should we solve this computational problem?
  - Exploit the structure of the search space and the cost function
  - That is, exploit decomposition of the scoring function

Approaches for inference

- Exact vs. approximate inference
  - Should the maximization be performed exactly?
    - Or is a close-to-highest-scoring structure good enough?
  - **Exact**: Search, dynamic programming, integer linear programming, ….
  - **Heuristic**: Gibbs sampling, belief propagation (some cases), beam search, linear programming relaxations (some cases), …
    - Also called *approximate inference*
- Randomized vs. deterministic
  - Relevant for approximate inference: If I run the inference program twice, will I get the same answer?

Coming up

- Graph algorithms, dynamic programing, greedy search
  - We have seen Viterbi algorithm

    Uses a cleverly defined ordering to decompose the output into a sequence of decisions

- Formulating general inference as integer linear programs
  - And variants of this idea

- Heuristics for inference
  - Sampling, Gibbs Sampling
  - Approximate graph search, beam search
  - Learning to search

Questions?