**Structured Learning**

Learning how to make joint predictions

<table>
<thead>
<tr>
<th>Task</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parsing</td>
<td>They operate ships and banks.</td>
<td>Read them.</td>
</tr>
<tr>
<td>Segmentation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Structured Learning Algorithms**

- Structured Perceptron Training
- Loop until stopping condition is met:
  
  For each \((x_i, y_i)\) pair:
  
  \[
  y = \arg \max_{y} w^T \phi(x_i, y) + \eta \]
  
  \[
  w \leftarrow w + \eta \nabla \phi(x_i, y) \]  
  
  \(\eta\): learning rate
  
  - Structured SVMs

**Learning with Amortization**

- Many inference problems share the same solution
- Models converge after a few iterations.
- Exploit this redundancy by caching old inferences

**General Inference Framework**

- Formulating the inference as an Integer Linear Programming (ILP) \((\text{Roth} \& \text{Yoh} \ 04)\)
  
  \[
  \max \sum_{c \in C} y_c \delta y_c \leq h, y_c \in \{0, 1\} \]

- Inference using ILP has been successful in NLP & Vision tasks
- Dependency Parsing, Sentence Compression
- Any MPE problem w.r.t. any probabilistic model, can be formulated as an ILP \(\text{Roth} \& \text{Yoh} \ 04, \text{Sonntag} \ 10\)
- The inferences can be solved by any approach

**Approximate Amortized Inference**

- Approximate inference by relaxing the condition.
- **Theorem**: If the following conditions are satisfied
  
  1. Same # variables & same constraints (same equivalence class)
  2. \(\forall i, 2(x_{ij}^* - 1)(c_{ij} - c_{ij}) \geq -c_{ij}\) then \(x^*_{ij}\) is a \(\frac{1}{2(c_{ij} - c_{ij})}\)-approximate solution to Q (M: a constant)

- DCD with fixed \(\epsilon\)
- Inference with an undegenerating approximate method
- DCD algorithm will stop and the empirical risk of the trained model is bounded (related to Finley & Joachims 08)
- DCD with adaptive \(\epsilon\)
- Guarantees to return an exact model