Lecture 4:
First look at structures

Kai-Wei Chang
CS @ University of Virginia
kw@kwchang.net

Some slides are adapted from Vivek Skirmar and Dan Roth
Previous Lecture

- Binary linear classification
  - Perceptron, SVMs, Logistic regression, Naïve Bayes
  - Output: $y \in \{1, -1\}$
- Multi-class classification
  - Multiclass Perceptron, Multiclass SVM…
  - Output: $y \in \{1, 2, 3, \ldots, K\}$

Question?
What we have seen: multiclass

- Reducing multiclass to binary
  - One-against-all & One-vs-one
  - Error correcting codes
  - Extension: Learning to search

- Training a single classifier
  - Multiclass Perceptron: Kesler’s construction
  - Multiclass SVMs: Crammer&Singer formulation
  - Multinomial logistic regression
  - Extension: Graphical models
This lecture

- What is structured output?
- Multiclass as structure
- Sequence as structure
- Next lecture: general graph structure
Global decisions

- “Understanding” is a global decision
  - Several local decisions play a role
  - There are mutual dependencies on their outcome.
- Essential to make coherent decisions
  - Joint, Global Inference
Inference with Constraints
[Roth&Yih’04,07,....]

Bernie’s wife, Jane, is a native of Brooklyn

E₁  →  E₂  →  E₃

R₁₂  R₂₃

Models could be learned separately/jointly; constraints may come up only at decision time.
Inference with Constraints [Roth&Yih’04,07,...]

Models could be learned separately/jointly; constraints may come up only at decision time.
Bernie’s wife, Jane, is a native of Brooklyn

Models could be learned separately/jointly; constraints may come up only at decision time.
Semantic Role Labeling

- To predict arguments of verbs

The bus was **heading** for Nairobi in Kenya

- **Predicate**
  - **Relation**: Head
  - **Mover**[A0]: the bus
  - **Destination**[A1]: Nairobi in Kenya

*Example from Vivek Srikumar*
How to?

The bus was **heading** for Nairobi in Kenya.

Identify

classify

Score candidate verb/arguments and their correlations

Inference
The bus was **heading** for Nairobi in Kenya.

Special label, meaning “Not an argument”
The bus was **heading** for Nairobi in Kenya.

```
0.1
0.5
0.2
0.1
0.1

Special label, meaning “Not an argument”

0.4
0.1
0.1
0.1
0.3

Total: **2.0**
```

heading (**The bus**, for Nairobi, for Nairobi in Kenya)

---

Computer Science at the University of Virginia
The bus was **heading** for Nairobi in Kenya.
Structured output is...

- A predefine structure

<table>
<thead>
<tr>
<th>Predicate</th>
<th>A0</th>
<th>A1</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head</td>
<td>The bus</td>
<td>Nairobi in Kenya</td>
<td>-</td>
</tr>
</tbody>
</table>

- Can be represented as a graph

```
   Head
  /   \
A0  A1

The bus  Nairobi in Kenya
```
Prediction result of a trained HMM


Violates lots of natural constraints!
We can fix it by adding constraints

Sequential tagging

- The process of assigning a part-of-speech to each word in a collection (sentence).

<table>
<thead>
<tr>
<th>WORD</th>
<th>tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>koala</td>
<td>N</td>
</tr>
<tr>
<td>put</td>
<td>V</td>
</tr>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>keys</td>
<td>N</td>
</tr>
<tr>
<td>on</td>
<td>P</td>
</tr>
<tr>
<td>the</td>
<td>DET</td>
</tr>
<tr>
<td>table</td>
<td>N</td>
</tr>
</tbody>
</table>
Let’s try

a dog chases a cat.

a fox 9 us

a boy 3 us

What is the POS tag sequence of the following sentence?

Don’t worry! There is no problem with your eyes or computer.
Let’s try

- a/DT dog/NN is/VBZ chasing/VBG a/DT cat/NN ./
- a/DT fox/NN is/VBZ running/VBG ./
- a/DT boy/NN is/VBZ singing/VBG ./
- a/DT happy/JJ bird/NN
- a happy cat was singing .
How you predict the tags?

- Two types of information are useful
  - Relations between words and tags
  - Relations between tags and tags
    - DT NN, DT JJ NN...
    - Fed in “The Fed” is a Noun because it follows a Determiner

- One possible model
  - Each output label is dependent on its neighbors in addition to the input
HMM

\[ y^* = \arg\max_{y \in \mathcal{Y}} P(y_0)P(x_0|y_0) \prod_{i=1}^{n-1} P(y_i|y_{i-1})P(x_i|y_i) \]

\[ P(y_0) \quad P(y_1|y_0) \quad P(y_2|y_1) \quad P(y_3|y_2) \quad P(y_4|y_3) \]

\[ P(x_0|y_0) \quad P(x_1|y_1) \quad P(x_2|y_2) \quad P(x_3|y_3) \quad P(x_4|y_4) \]
$P(y_0)$  $P(y_1 | y_0)$  $P(y_2 | y_1)$  $P(y_3 | y_2)$  $P(y_4 | y_3)$

$P(x_0 | y_0)$  $P(x_1 | y_1)$  $P(x_2 | y_2)$  $P(x_3 | y_3)$  $P(x_4 | y_4)$

Example: the man saw the dog
What is structured output?

- A graph, possibly labeled and/or directed
  - Possibly from a restricted family, such as chains, trees, etc.
- A discrete representation of input/output
- A collection of inter-dependent decisions
- E.g. A table, the SRL frame output, a sequence of labels, etc.
Combinatorial optimization problem

\[ \hat{y} = \arg\max_{y \in \mathcal{Y}} f(y; w, x) \]

- **Inference/Test**: given \( w, x \), solve \( \arg\max \)
- **Learning/Training**: find a good \( w \)
Challenges with structured output

- We cannot train a separate weight vector for each possible inference outcome (why?)
  - For multi-class we train one weight vector for each class

- We cannot enumerate all possible structures for inference
  - Inference for multiclass was easy
Deal with combinatorial output

- Decompose the output into parts that are labeled
- Define a graph to represent
  - how the parts interact with each other
  - These labeled interacting parts are scored; the total score for the graph is the sum of scores of each part
  - an inference algorithm to assign labels to all the parts
Decomposing the output: example

Note: The output $y$ is a labeled assignment of the nodes and edges

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

Goal: Find the highest scoring labeling such that the edges that are colored form a tree

The input $x$ not shown here
Decomposing the output: example

One option: Decompose fully. All nodes and edges are independently scored

score(x, y) = \sum_{n \in \text{nodes}(x,y)} \text{score}(n) + \sum_{e \in \text{edges}(x,y)} \text{score}(e)
Decomposing the output: example

**One option: Decompose fully.** All nodes and edges are independently scored

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

\[
\text{score}(x, y) = \sum_{n \in \text{nodes}(x, y)} \text{score}(n) + \sum_{e \in \text{edges}(x, y)} \text{score}(e)
\]

**Prediction:** \(\arg\max_y \text{score}(x, y)\) s.t. \(y\) forms a tree
Decomposing the output: example

Another possibility: Score each edge and its nodes together

And many other edges...

Each patch represents piece that is scored independently

$$\text{score}(x, y) = \sum_{n_1, n_2 \in \text{nodes}(x, y)} \sum_{e \in \text{edges}(x, y)} \text{score}(n_1, n_2, e)$$
Agreement describes the structure

Invalid! Two parts disagree on the label for this node
Multi-class as structured output

- A structure is...
  - A graph (in general, hypergraph), possibly labeled and/or directed
  - A collection of interdependent decisions
  - The output of a combinatorial optimization problem
    \[
    \text{argmax}_{y \in \text{all outputs}} \text{score}(x, y)
    \]

- Multiclass
  - A graph with one node and no edges
    - Node label is the output
  - Can be composed via multiple decisions
  - Winner-take-all
    \[
    \text{argmax}_i w^T \phi(x, i)
    \]
Another decomposition idea

- A structure is...
  - A graph (in general, hypergraph), possibly labeled and/or directed
  - A collection of inter-dependent decisions
  - The output of a combinatorial optimization problem
    \[ \arg\max_{y \in \text{all outputs}} \text{score}(x, y) \]

- Multiclass
  - A set of binary variables
  - Only one node can be on
Inference

❖ Each part is scored **independently**
   ❖ **Key observation**: Number of possible inference outcomes for each part may not be large •
   ❖ Even if the number of possible structures might be large

❖ **Inference**: How to glue together the pieces to build a valid output?
   ❖ Depends on the “shape” of the output
   ❖ Efficient algorithms may exist
   ❖ General algorithms: Integer linear programming
Recap: Key questions

**Modeling:** How to decompose the structure?

**Inference:** How to obtain the best coherent output?

**Training:** Learn the scoring function to give part scores

\[
\text{score}(x, y) = \sum_{n \in \text{nodes}(x, y)} \text{score}(n) + \sum_{e \in \text{edges}(x, y)} \text{score}(e)
\]
Key questions in structured prediction

Model definition
What are the parts of the output?
What are the inter-dependencies?

Background knowledge about domain
Key questions in structured prediction

Model definition
What are the parts of the output?
What are the inter-dependencies?

Background knowledge about domain

How to do inference?
Key questions in structured prediction

Model definition
What are the parts of the output?
What are the inter-dependencies?

How to train the model?
Background knowledge about domain
How to do inference?
Key questions in structured prediction

- **Model definition**
  What are the parts of the output?
  What are the inter-dependencies?

- **Data** annotation difficulty

- **How to train** the model?

- **Background knowledge** about domain

- **How to do inference**?

---

lecture 4: First look at structures
Key questions in structured prediction

- **Data annotation difficulty**
- **Model definition**
  - What are the parts of the output?
  - What are the inter-dependencies?
- **How to train the model?**
- **Background knowledge about domain**
- **How to do inference?**
- **Semi-supervised/indirectly supervised?**
Next step

- Sequential tagging problems
  - Modeling/ training/ inference

- Other general structured prediction problems
  - Modeling/ training/ inference