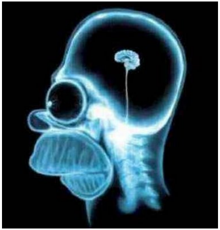
Seminar in Algorithms for NLP (Structured Prediction)

Yoav Goldberg
yogo@macs.biu.ac.il

March 5, 2013

Natural Language Processing?

NLP

text →  → meaning

Reminder: Machine Learning

(supervised) Machine Learning

Input

(Labeled) Data

Output

Function $f(x)$

(supervised) Machine Learning

Input

(Labeled) Data

oranges

apples

Output

Function $f(x)$

(supervised) Machine Learning

Input

(Labeled) Data

oranges

apples

Output

Function $f(x)$

f()=`orange`

f()=`apple`

(supervised) Machine Learning

Input

(Labeled) Data

oranges

apples

Output

Function $f(x)$

f()=`orange`

f()=`apple`

f()=

(supervised) Machine Learning

Input

(Labeled) Data

oranges

apples

Output

Function $f(x)$

f()=`orange`

f()=`apple`

f()=`apple`

Representation

Functions return numbers

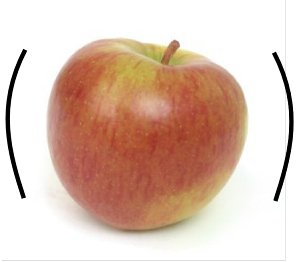
$f(x) > 0 \rightarrow$ `apple`

$f(x) < 0 \rightarrow$ `orange`

Representation

f() ?

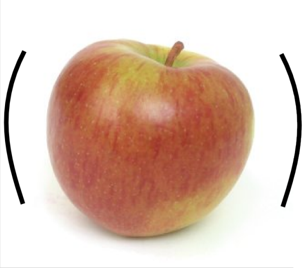
Representation

$\phi\Big($  $\Big)$

$=$

(4.2, 212, 3.4, 1332)

Representation

$\phi\Bigg($  $\Bigg)$

$=$

$$(\underbrace{0, 1, 0, 0}_{\text{diameter}}, \underbrace{0, 1, 0}_{\text{weight}}, \underbrace{1, 0, 0}_{\text{softness}}, \underbrace{0, 0, 1}_{\text{color}})$$

3.5 to 4

4 to 4.5

4.5 to 5

...

Learning Linear Functions

f(<1,0,0,0,1,0,0,1,1,0,0,0,1>)

Learning Linear Functions

f(<1,0,0,0,1,0,0,1,1,0,0,0,1>)

$$f(x) = wx$$

Learning Linear Functions

f(<1,0,0,0,1,0,0,1,1,0,0,0,1>)

$$f(x) = wx$$

$$f(x) = w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots + w_n x_n$$

Learning Linear Functions

f(<1,0,0,0,1,0,0,1,1,0,0,0,1>)

$$f(x) = wx$$

$$f(x) = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n$$

Learning: find *w* that classifies well
(separates apples from oranges)

Learning Linear Functions

f(<1,0,0,0,1,0,0,1,1,0,0,0,1>)

$$f(x) = wx$$

$$f(x) = w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots + w_n x_n$$

Learning: find *w* that classifies well
(separates apples from oranges)

Many algorithms (MaxEnt, SVM, ...)

Types of learning problems

Goal of Learning

Given instances x_i and labels $y_i \in \mathcal{Y}$, learn a function $f(x)$ such that, on most inputs x_i, $f(x_i) = y_i$, and which will generalize well to unseen (x, y) pairs.

(not quite accurate: more formally we want $f()$ to achieve *low loss* with respect to some *loss function*, under *regularization constraints*.)

Common learning scenarios

- Binary Classification: $\mathcal{Y} = \{-1, 1\}$
- Multiclass Classification: $\mathcal{Y} = \{0, 1, \ldots, k\}$
- Regression: $\mathcal{Y} = \mathbb{R}$

The Perceptron Algorithm (binary)

Inputs: items $x_1, \ldots, x_n$, classes $y_1, \ldots, y_n$, feature function $\phi(x)$

$\mathbf{w} \leftarrow 0$

for k iterations **do**

  for x_i, y_i **do**

    $y' \leftarrow sign(\mathbf{w} \cdot \phi(x_i))$

    if $y' \neq y_i$ **then**

      $\mathbf{w} \leftarrow \mathbf{w} + y_i \phi(x_i)$

return w

The Perceptron Algorithm (multiclass)

Inputs: items $x_1, \ldots, x_n$, classes $y_1, \ldots, y_n$, feature function $\phi(x, y)$

$\mathbf{w} \leftarrow 0$

for k iterations **do**

  for x_i, y_i **do**

    $y' \leftarrow argmax_y(\mathbf{w} \cdot \phi(x_i, y))$

    if $y' \neq y_i$ **then**

      $\mathbf{w} \leftarrow \mathbf{w} + \phi(x_i, y_i) - \phi(x_i, y')$

return w

Structured Prediction:

Predicting complex outputs

Structured Prediction

Sequence Tagging

The boy in the bright blue jeans jumped up on the stage

↓

`DT NN PREP DT ADJ ADJ NN VB PRT PREP DT NN`

Structured Prediction

Sequence Segmentation - Chunking

The boy in the bright blue jeans jumped up on the stage

↓

`[The boy] in [the bright blue jeans] [jumped up] on [the stage]`

Structured Prediction

Sequence Segmentation - Named Entities

Donald Trump will endorse Mitt Romney in Las Vegas this Thursday.

↓

Donald Trump will endorse Mitt Romney in Las Vegas this Thursday

Structured Prediction

Sequence Segmentation - Named Entities

Donald Trump will endorse Mitt Romney in Las Vegas this Thursday.

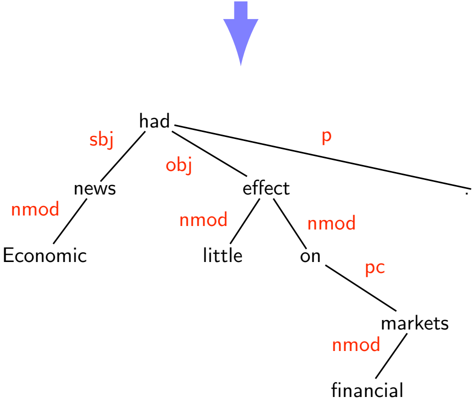
Donald Trump will endorse Mitt Romney in Las Vegas this Thursday

(Sequence Segmentation is a special form of a tagging problem)

Structured Prediction

Syntactic Parsing

Economic news had little effect on financial markets .



Structured Prediction

Sentence Simplification

Economic news had little effect on financial markets

↓

news had little effect on markets

Structured Prediction

Sentence Simplification

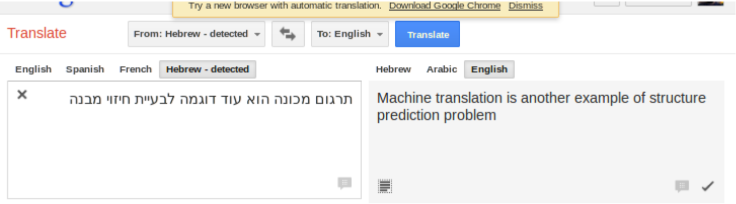
Economic news had little effect on financial markets

↓

news had effect on markets

Structured Prediction

String Translation

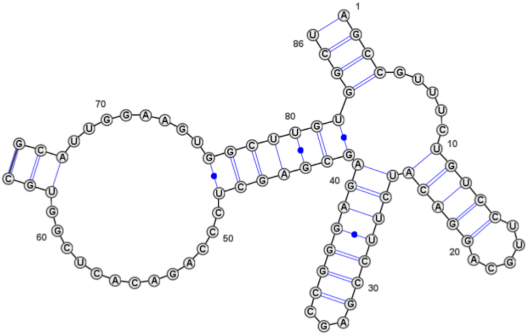


Structured Prediction

RNA Folding

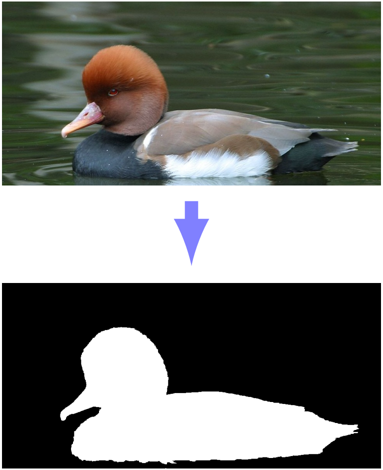
AGCCGUUUCUGUCCUUGCAGGACAUCUUCCGAGCCGGGAGAGCGAGCUCCCAGACACUCGGUGCGCAUUGGAAGUGGCUUGUGGCU

((((.....(((((.....)))))(((((((...)))))))(((((((.............((()))........)))))))))))



Structured Prediction

Image Segmentation



Structured Prediction:

Predicting complex outputs

Structured Prediction:

Predicting complex outputs

Predicting **interesting** outputs

Structured Prediction

Output space is large
(k^n possible sequences for sentence of length n)

Output space is constrained
("output must be a projective tree")

Many correlated decisions
(labels can depend on other labels)

How To Solve

Ignore Correlations?

- Treat as multiple independent classification problems.
- Solve each one individually.

Good

- Very fast (linear time prediction)

Bad

- Ignores the structure of the output space.
- Hard to encode constraints (easy for sequences, hard for trees)
- Does not perform well.

How to Solve example/reminder – HMM and Viterbi

probability of a tag sequence:

$$P(w_1, \ldots, w_n, t_1, \ldots, t_n) = \prod_{i=1}^{N} p(t_i | t_{i-1}) p(w_i | t_i)$$

HMM has two sets of parameters:

$$t(t1, t2) = p(t2 | t1) \qquad e(t, w) = p(w | t)$$

based on these, we define a score:

$$s(t1, t2, w) = t(t1, t2) e(t, w)$$

```
Initialize D(0,START) = 0
for i in 1 to n do
    for t ∈ tags do
        D(i, t_j) = max_{t'∈tags}(D(i − 1, t') × s(t', t, w_i))
```

- Can we do better than HMM for tagging?
- How do we generalize beyond sequence tagging?