R1 One vs All/Rest — $\arg\max_i h_i(x,y)$

$k(k-1)/2$ AvA R2

$k \begin{cases} y_1 = \{1, \neq 1\} \\ y_2 = \{2, \neq 2\} \\ \dots \end{cases}$

$y = \{1, \dots, k\}$

Reduction

Generalization

$h_k(x)$

$h_j(x) - h_i(x)$, $i \neq j$

Multiclass & Structured Prediction

Binary classification

- Perceptron $\{0, 1\}$
- SVM $\{-1, 1\}$

- Logistic Regression
  - Multinomial

DS-GA 1003 Machine Learning

CDS, NYU

March 31, 2021

Pros

- corrections
- no gradient

- Margin based
- Sparse solution in span of support vectors

- probabilities
- Max likelihood Estimation

Cons

not linearly separable

no probabilities

[JK] Annotated

Multiclass Hypothesis Space: Reframed

- **General [Discrete] Output Space:** $\mathcal{Y} = \{1, \ldots, k\}$

Multiclass: Perceptron, SVM

- **Base Hypothesis Space:** $\mathcal{H} = \{h : \mathcal{X} \times \mathcal{Y} \to \mathrm{R}\}$
    - $h(x, y)$ gives **compatibility score** between input x and output y

or energy

- **Multiclass Hypothesis Space**

$$\mathcal{F} = \left\{ x \mapsto \arg\max_{y \in \mathcal{Y}} h(x, y) \mid h \in \mathcal{H} \right\}$$

- Final prediction function is an $f \in \mathcal{F}$.
- For each $f \in \mathcal{F}$ there is an underlying compatibility score function $h \in \mathcal{H}$.

Part-of-speech (POS) Tagging

Structured Prediction
- Sequences → NLP, DNA/seq
- Trees, graphs

- Given a sentence, give a part of speech tag for each word:

x	$\underbrace{\text{[START]}}_{x_0}$	$\underbrace{\text{He}}_{x_1}$	$\underbrace{\text{eats}}_{x_2}$	$\underbrace{\text{apples}}_{x_3}$
y	$\underbrace{\text{[START]}}_{y_0}$	$\underbrace{\text{Pronoun}}_{y_1}$	$\underbrace{\text{Verb}}_{y_2}$	$\underbrace{\text{Noun}}_{y_3}$

- $\mathcal{V} = \{\text{all English words}\} \cup \{\text{[START]}, \text{"."}\}$ — Vocabulary
- $\mathcal{P} = \{\text{START}, \text{Pronoun}, \text{Verb}, \text{Noun}, \text{Adjective}\}$ — POS
- $\mathcal{X} = \mathcal{V}^n$, $n = 1, 2, 3, \ldots$ [Word sequences of any length]
- $\mathcal{Y} = \mathcal{P}^n$, $n = 1, 2, 3, \ldots$[Part of speech sequence of any length]

Structured Prediction

- A **structured prediction** problem is a multiclass problem in which $\mathcal{Y}$ is very large, but has (or we assume it has) a certain structure.
- For POS tagging, $\mathcal{Y}$ grows exponentially in the length of the sentence.
- Typical **structure** assumption: The POS labels form a Markov chain.
  - i.e. $y_{n+1} \mid y_n, y_{n-1}, \ldots, y_0$ is the same as $y_{n+1} \mid y_n$.

Markov Assumption

Local Feature Functions: Type 1 — Unary

- A "type 1" **local feature** only depends on
  - the label at a single position, say y_i (label of the ith word) and
  - x at any position
- Example: <s> He eats apples <s> He runs fast

  0 1 2

$$
\begin{aligned}
\phi_1(i,x,y_i) &= 1(x_i = \text{runs})1(y_i = \text{Verb}) \\
\phi_2(i,x,y_i) &= 1(x_i = \text{runs})1(y_i = \text{Noun}) \\
\phi_3(i,x,y_i) &= 1(x_{i-1} = \text{He})1(x_i = \text{runs})1(y_i = \text{Verb}) \\
&\vdots
\end{aligned}
$$

$$\Phi(i,x,y_i) = \left(\phi_1(i,x,y_i),\ \phi_2(i,x,y_i),\ \phi_3(i,x,y_i)\right)$$

Noun, Verb

$$\text{e.g. } \Phi(2,x,\boxed{y_2}) = \left(\begin{matrix}0\\0\end{matrix},\ \begin{matrix}0\\1\end{matrix},\ \begin{matrix}0\\0\end{matrix}\right)$$

Local Feature Functions: Type 2 — Markov

- A "type 2" **local feature** only depends on
  - the labels at 2 consecutive positions: y_{i-1} and y_i
  - x at any position
- Example:

$$
\begin{aligned}
\theta_1(i, x, y_{i-1}, y_i) &= 1(y_{i-1} = \text{Pronoun})1(y_i = \text{Verb}) \\
\theta_2(i, x, y_{i-1}, y_i) &= 1(y_{i-1} = \text{Pronoun})1(y_i = \text{Noun})
\end{aligned}
$$

$\vdots$

$\Theta(2, x, \text{Noun}) = (0, 1)$

Local Feature Vector and Compatibility Score

- At each position i in sequence, define the **local feature vector**: (class/label dependent)

$$
\begin{aligned}
\Psi_i(x, y_{i-1}, y_i) \quad = \quad & \overbrace{(\phi_1(i, x, y_i), \phi_2(i, x, y_i), \ldots,}^{\text{Type 1}} \\
& \underbrace{\theta_1(i, x, y_{i-1}, y_i), \theta_2(i, x, y_{i-1}, y_i), \ldots}_{\text{Type 2}})
\end{aligned}
$$

- **Local compatibility score** for (x, y) at position i is $\langle w, \Psi_i(x, y_{i-1}, y_i) \rangle$.

$h : \mathcal{X} \to \mathbb{R}$ $\qquad$ $\langle w^T \Psi_i \rangle$

Sequence Compatibility Score

- The **compatibility score** for the pair of sequences (x, y) is the sum of the local compatibility scores:

$$
\begin{aligned}
& \sum_i \langle w, \Psi_i(x, y_{i-1}, y_i) \rangle \\
= & \left\langle w, \sum_i \Psi_i(x, y_{i-1}, y_i) \right\rangle \\
= & \langle w, \Psi(x, y) \rangle, \quad - h(x)
\end{aligned}
$$

where we define the sequence feature vector by

$$\Psi(x, y) = \sum_i \Psi_i(x, y_{i-1}, y_i).$$

- So we see this is a special case of linear multiclass prediction.

Sequence Target Loss

$\Phi(x,y) \longmapsto h(x,y) = \langle w, \Phi(x,y) \rangle$

$f \in \{ x \mapsto \arg\max_{y \in \mathcal{Y}} h \}$

- How do we assess the loss for prediction sequence y' for example (x, y)?

$y' = \arg\max_{y \in \mathcal{Y}} h(x,y)$

- **Hamming loss** is common:

$$\Delta(y, y') = \frac{1}{|y|} \sum_{i=1}^{|y|} 1(y_i \neq y'_i)$$

e.g.

He	eats	apples
P	V	N
P	N	N
0	1	0

$= 1/3$

- Could generalize this as

$$\Delta(y, y') = \frac{1}{|y|} \sum_{i=1}^{|y|} \delta(y_i, y'_i)$$

- 0-1 loss → Perceptron
- Hinge loss → SVM

What remains to be done?

- To compute predictions, we need to find

$$\boxed{\underbrace{\arg\max_{y \in \mathcal{Y}} \langle w, \Psi(x,y) \rangle}_{\text{learnt}}}.$$

- This is straightforward for $|\mathcal{Y}|$ small.
- Now $|\mathcal{Y}|$ is exponentially large.
- Because Ψ breaks down into local functions only depending on 2 adjacent labels,
  - we can solve this efficiently using dynamic programming.
  - (Similar to Viterbi decoding.)

→ reduces time complexity from exponential to polynomial

- Learning can be done with SGD and a similar dynamic program.

References

- DS-GA 1003 Machine Learning Spring 2019