Bone vs All /Rest - argmax hilac,y) k(k-1)/2 Auf (2) 7 = 21, +19. Reduction
4 = 22, +29. hu(x) h; (a) - h; (a) Multiclass & Structured Prediction Binarry classification DS-GA 1003 Machine Learning Pros CDS, NYU to gradient actions actions + Perception 80,13 -+ SVM - Logistic Regression f Multinomial + Mare Weelihood Estimation [JK] Annotated DS-GA 1003 Machine Learning (CDS, NYU) March 31, 2021 Recitation 9 1/11

Multiclass Hypothesis Space: Reframed

- General [Discrete] Output Space: $y = \{1, ..., k\}$ Base Hypothesis Space: $\mathcal{H} = \{h : \mathcal{X} \times \mathcal{Y} \to R\}$

- - h(x,y) gives **compatibility score** between input x and output y
- 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/ey,

+ Trees, graphs

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

X	[START]	He	eats	apples
	<i>x</i> ₀	x_1	X2	<i>X</i> 3
y	[START]	Pronoun	Verb	Noun
	Vo	y_1	_ <u>y2</u>	<i>y</i> ₃
	70		1	

- V = {all English words}∪{[START],"."} bo cabulary
- $\mathcal{P} = \{START, Pronoun, Verb, Noun, Adjective\}$ Pos
- $\mathfrak{X} = (V^n)$ n = 1, 2, 3, ... [Word sequences of any length]
- $y = \mathcal{P}^n$, n = 1, 2, 3, ...[Part of speech sequence of any length]

Structured Prediction

- A structured prediction problem is a <u>multiclass problem</u> in which <u>y</u> is very large, but has (or we assume it has) a certain structure.
- For POS tagging, y grows exponentially in the length of the sentence.

Typical structure assumption: The POS labels form a Markov chain.

• i.e. $y_{n+1} | y_n, y_{n-1}, ..., y_0$ is the same as $y_{n+1} | y_n, y_{n-1} | y_n |$



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: 45 the east apples 45th runs fast $\phi_1(i, x, v_i) = 1(x_i = \text{runs})1(v_i = \text{Verb})$ $\Phi_2(i, x, y_i) = 1(x_i = \text{runs})1(y_i = \text{Noun})$ $\phi_3(i, x, v_i) = 1(x_{i-1} = \text{He})1(x_i = \text{runs})1(v_i = \text{Verb})$ Φ(i, x, yi) = (θ, (i, x, yi), θ, (i, x, yi))
 e.g. Φ(z, x, yi) = (θ, (i, x, yi), θ, (i, x, yi))

Local Feature Functions: Type 2 — Markor

- 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:

$$\begin{array}{rcl} \theta_1(i,x,y_{i-1},y_i) &=& 1(y_{i-1}=\operatorname{Pronoun})1(y_i=\operatorname{Verb}) \\ \theta_2(i,x,y_{i-1},y_i) &=& 1(y_{i-1}=\operatorname{Pronoun})1(y_i=\operatorname{Noun}) \\ \vdots &&& \\ \Theta\left(2,y_{i},y_{i}\right) &=& \left(0,y_{i}\right) \end{array}$$

Local Feature Vector and Compatibility Score

• At each position i in sequence, define the local feature vector: (also hale)

$$\Psi_i(x,y_{i-1},y_i) = (\phi_1(i,x,y_i),\phi_2(i,x,y_i),...,\phi_1(i,x,y_{i-1},y_i),\theta_2(i,x,y_{i-1},y_i),\cdots)$$

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

Sequence Compatibility Score

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

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

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

prediction sequence
$$y'$$
 for example (x,y) ?

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

Could generalize this as

$$\Delta(y,y') = \frac{1}{|y|} \sum_{i=1}^{|y|} \underbrace{\delta(y_i,y_i')}_{\text{Uinge loss}}$$

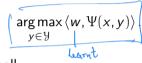
$$= \frac{1}{2}$$

$$+ \frac{1}{|y|} \sum_{i=1}^{|y|} \underbrace{\delta(y_i,y_i')}_{\text{Uinge loss}}$$

$$= \frac{1}{2}$$

What remains to be done?

To compute predictions, we need to find



- This is straightforward for |y| small.
- Now |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.)

treduces time complexity from exponential to

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

References

• DS-GA 1003 Machine Learning Spring 2019