Recitation 7

MultiClass Structured Prediction

Colin

Spring 2022

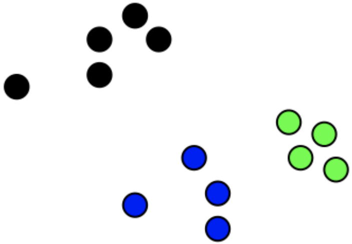
Mar 30

Outline

- Logistics
- Types of Multi-class Classification Approaches
  - One VS All
  - All VS All (all pairs)
  - Multi-class

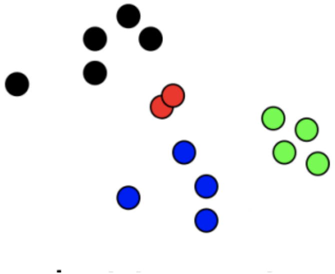
One VS All

- Very simple idea, fit a classifier for every class
- Strong assumption of linearly separable
  - Not the case for most of the problems
- Highest score wins



All VS All

- One solution to One VS All's linearly separable assumption
- Train nC_2 classifiers
- Majority votes
- Extremely high cost when number of classes is large



ECOC

- One solution to All VS All's large computation cost
- Train less classifiers
- Perform another mapping on its results
- Introduces the concept of latent variable, can be viewed as feature mapping

Multi-class

- Suppose we want to use one linear boundary, how can we adjust the setup?

$$w = \left(\underbrace{-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}}_{w_1}, \underbrace{0, 1}_{w_2}, \underbrace{\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}}_{w_3} \right)$$

$$\begin{aligned} \psi(x,1) &:= (x_1, x_2, 0, 0, 0, 0) \\ \psi(x,2) &:= (0, 0, x_1, x_2, 0, 0) \\ \psi(x,3) &:= (0, 0, 0, 0, x_1, x_2) \end{aligned}$$

Note in this case, the input to our feature map is x and y.

Review

Final step: Adjustments

- How can we make it differentiable
- Loss functions
  - Hamming Loss

$$y'_i = 1(y'_i = \max_j y'_j)$$

$$l(y, y') = \frac{1}{k} \sum_i^k 1(y_i \neq y'_i)$$

Final step: Adjustments

- Hamming loss is not differentiable due to the max operator
  - Softmax and cross entropy function:

$$
\begin{aligned}
z &= f(x) \quad \text{Note } z_i \in \mathbb{R} \\
y_i' &= \frac{e^{z_i}}{\sum_j e^{z_j}} \\
l(y, y') &= -\sum_i y_i log(y_i') \\
&= log\left(\frac{e^{f(x)_i}}{\sum_j e^{f(x)_j}}\right) \quad \text{where } y_i = 1
\end{aligned}
$$

  - This is analogical to the logistic loss

Multiclass Hypothesis Space: Reframed

- **General [Discrete] Output Space:** $\mathcal{Y}$
- **Base Hypothesis Space:** $\mathcal{H} = \{h : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}\}$
  - $h(x, y)$ 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

- 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	$\underbrace{\text{[START]}}_{y_0}$	$\underbrace{\text{Pronoun}}_{y_1}$	$\underbrace{\text{Verb}}_{y_2}$	$\underbrace{\text{Noun}}_{y_3}$

- $\mathcal{V} = \{\text{all English words}\} \cup \{\text{[START]}, \text{"."}\}$
- $\mathcal{P} = \{\text{START}, \text{Pronoun,Verb,Noun,Adjective}\}$
- $\mathcal{X} = \mathcal{V}^n$, $n = 1, 2, 3, \ldots$ [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 $y_{n+1} \mid y_n$.

Local Feature Functions: Type 1

- 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:

$$
\begin{aligned}
\varphi_1(i, x, y_i) &= \mathbf{1}(x_i = runs)\mathbf{1}(y_i = Verb) \\
\varphi_2(i, x, y_i) &= \mathbf{1}(x_i = runs)\mathbf{1}(y_i = Noun) \\
\varphi_3(i, x, y_i) &= \mathbf{1}(x_{i-1} = He)\mathbf{1}(x_i = runs)\mathbf{1}(y_i = Verb)
\end{aligned}
$$

Local Feature Functions: Type 2

- 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:

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

Local Feature Vector and Compatibility Score

- At each position i in sequence, define the **local feature vector**:

$$\begin{aligned}\Psi_i(x, y_{i-1}, y_i) &= (\varphi_1(i, x, y_i), \varphi_2(i, x, y_i), \ldots, \\ &\quad \theta_1(i, x, y_{i-1}, y_i), \theta_2(i, x, y_{i-1}, y_i), \ldots)\end{aligned}$$

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

Sequence Compatibility Score

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

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

  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

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

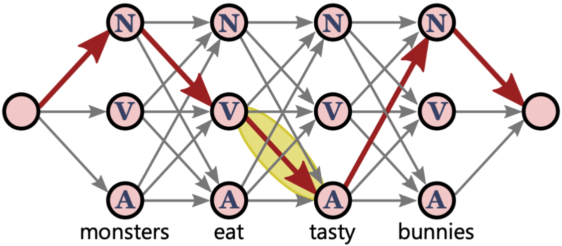
$$\Delta(y, y') = \frac{1}{|y|} \sum_{i=1}^{|y|} \mathbf{1} y_i \neq y'_i$$

- Could generalize this as

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

The argmax problem for sequences

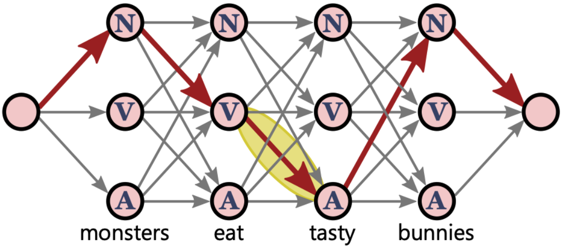
Problem To compute predictions, we need to find $\arg\max_{y \in \mathcal{Y}(x)} \langle w, \Psi(x, y) \rangle$, and $|\mathcal{Y}(x)|$ is exponentially large.



The argmax problem for sequences

Problem: To compute predictions, we need to find $\arg\max_{y \in \mathcal{Y}(x)} \langle w, \Psi(x, y) \rangle$, and $|\mathcal{Y}(x)|$ is exponentially large.

Observation: $\Psi(x, y)$ decomposes to $\sum_i \Psi_i(x, y)$.

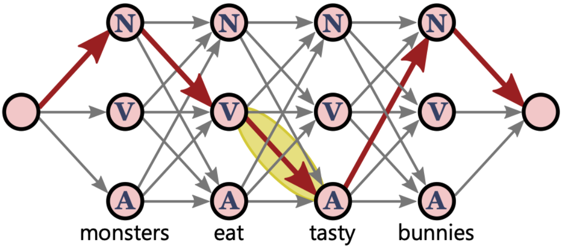


The argmax problem for sequences

Problem To compute predictions, we need to find $\arg\max_{y \in \mathcal{Y}(x)} \langle w, \Psi(x,y) \rangle$, and $|\mathcal{Y}(x)|$ is exponentially large.

Observation $\Psi(x,y)$ decomposes to $\sum_i \Psi_i(x,y)$.

Solution Dynamic programming (similar to the Viterbi algorithm)



References

- DS-GA 1003 Machine Learning Spring 2019