Structured Perceptrons
Algorithm, Viterbi, and a Worked Example

James Pustejovsky

CS115B
Brandeis University

February 4, 2025

Outline

Structured Perceptron Setup

Structured Prediction Setting

- $\mathbf{x}$: input (e.g., a sentence $(x_1, x_2, \ldots, x_n)$).
- $\mathbf{y}$: structured output (e.g., a sequence of POS tags $(y_1, y_2, \ldots, y_n)$).
- $\Phi(\mathbf{x}, \mathbf{y})$: feature representation combining input $\mathbf{x}$ and output $\mathbf{y}$.
- Model score function:

$$S(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle.$$

- Prediction at test time:

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle,$$

where $\mathcal{Y}(\mathbf{x})$ is all feasible structures (tag sequences) for $\mathbf{x}$.

Structured Perceptron Algorithm (Online)

Algorithm 1 Structured Perceptron (Online Version)

Input: Training set $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, number of epochs T.
Initialize: $\mathbf{w}^{(0)} = \mathbf{0}$.
for $t = 1 \to T$ **do**
  for $i = 1 \to N$ **do**
    $\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \langle \mathbf{w}^{(t)}, \Phi(\mathbf{x}_i, \mathbf{y}) \rangle$
    if $\hat{\mathbf{y}} \neq \mathbf{y}_i$ **then**

$$\mathbf{w}^{(t)} \leftarrow \mathbf{w}^{(t)} + \left[\Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \hat{\mathbf{y}})\right]$$

    end if
  end for
end for
Return: $\mathbf{w}^{(T)}$.

- Each training step requires arg max inference, e.g. via Viterbi for sequence labeling.
- If the predicted structure $\hat{\mathbf{y}}$ differs from the gold $\mathbf{y}_i$, we update.

Feature Decomposition for Sequences

Sequence Labeling Setup:

$$\mathbf{x} = (x_1, \ldots, x_n), \quad \mathbf{y} = (y_1, \ldots, y_n).$$

Often, features decompose into:

$$\Phi(\mathbf{x}, \mathbf{y}) = \sum_{t=1}^{n} \phi_{\text{emit}}(x_t, y_t) + \sum_{t=2}^{n} \phi_{\text{trans}}(y_{t-1}, y_t).$$

- $\phi_{\text{emit}}(x_t, y_t)$: features linking the word x_t with its tag y_t.
- $\phi_{\text{trans}}(y_{t-1}, y_t)$: features for tag-to-tag transitions.
- Score function:

$$S(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \sum_{t=1}^{n} \langle \mathbf{w}, \phi_{\text{emit}}(x_t, y_t) \rangle + \sum_{t=2}^{n} \langle \mathbf{w}, \phi_{\text{trans}}(y_{t-1}, y_t) \rangle.$$

Viterbi Algorithm Formulas

Goal:

$$\hat{\mathbf{y}} = \arg \max_{y_1,\ldots,y_n} \left\{ \sum_{t=1}^{n} \langle \mathbf{w}, \phi_{\text{emit}}(x_t, y_t) \rangle + \sum_{t=2}^{n} \langle \mathbf{w}, \phi_{\text{trans}}(y_{t-1}, y_t) \rangle \right\}.$$

Viterbi DP:

$$\delta_t(u) = \max_{v \in \mathcal{T}} \left\{ \delta_{t-1}(v) \; + \; \langle \mathbf{w}, \phi_{\text{trans}}(v, u) \rangle \; + \; \langle \mathbf{w}, \phi_{\text{emit}}(x_t, u) \rangle \right\},$$

with

$$\delta_1(u) = \langle \mathbf{w}, \phi_{\text{emit}}(x_1, u) \rangle \quad \text{(for all } u \in \mathcal{T}\text{)}.$$

$$\hat{y}_n = \arg \max_{u \in \mathcal{T}} \delta_n(u), \quad \text{then backtrack to get } (\hat{y}_1, \ldots, \hat{y}_n).$$

Example Sentence and Tag Set

Sentence: $\mathbf{x} = [\text{“The”}, \text{“dog”}, \text{“ate”}, \text{“my”}, \text{“homework”}]$
Toy Tag Set $\mathcal{T}$:

$$\{\text{DET}, \text{NOUN}, \text{VERB}, \text{PRON}\}.$$

(We keep it small for illustration.)
Gold Tag Sequence ($\mathbf{y}$):

DET (The), NOUN (dog), VERB (ate), PRON (my), NOUN (homework).

Features (Simplified Example)

Feature representation for emission and transitions

$$\phi_{\text{emit}}(x_t, y_t) = \begin{cases} 1 & \text{if } (x_t = \text{“dog”} \wedge y_t = \text{NOUN}), \\ 1 & \text{if } (x_t = \text{“ate”} \wedge y_t = \text{VERB}), \\ 1 & \text{if } (x_t = \text{“my”} \wedge y_t = \text{PRON}), \\ 1 & \text{if } (x_t = \text{“The”} \wedge y_t = \text{DET}), \\ 1 & \text{if } (x_t = \text{“homework”} \wedge y_t = \text{NOUN}), \\ 0 & \text{otherwise.} \end{cases}$$

$$\phi_{\text{trans}}(y_{t-1}, y_t) = \begin{cases} 1 & \text{if } (y_{t-1} = \text{DET} \wedge y_t = \text{NOUN}), \\ 1 & \text{if } (y_{t-1} = \text{NOUN} \wedge y_t = \text{VERB}), \\ 1 & \text{if } (y_{t-1} = \text{VERB} \wedge y_t = \text{PRON}), \\ 1 & \text{if } (y_{t-1} = \text{PRON} \wedge y_t = \text{NOUN}), \\ 0 & \text{otherwise.} \end{cases}$$

Note: - Each non-zero feature has a corresponding weight in $\mathbf{w}$.

Viterbi Example (Partial Illustration)

Step 1 (Initialization):

$$\delta_1(\text{DET}) = \langle \mathbf{w}, \phi_{\text{emit}}(\text{“The”}, \text{DET}) \rangle,$$

$$\delta_1(\text{NOUN}) = \langle \mathbf{w}, \phi_{\text{emit}}(\text{“The”}, \text{NOUN}) \rangle,$$

$$\delta_1(\text{VERB}) = \ldots, \quad \delta_1(\text{PRON}) = \ldots$$

Only $\phi_{\text{emit}}(\text{“The”}, \text{DET})$ might be 1; others could be 0.

Step 2 (Recursion): For word “dog” at $t = 2$:

$$\delta_2(u) = \max_{v \in \mathcal{T}} \Big[\delta_1(v) + \langle \mathbf{w}, \phi_{\text{trans}}(v, u) \rangle + \langle \mathbf{w}, \phi_{\text{emit}}(\text{“dog”}, u) \rangle \Big].$$

Compute for $u \in \{\text{DET}, \text{NOUN}, \text{VERB}, \text{PRON}\}$.

Continue for “ate”, “my”, “homework” up to $t = 5$. Then pick:

$$\hat{y}_5 = \arg\max_u \delta_5(u),$$

and backtrack to get $\hat{y}_1, \ldots, \hat{y}_5$.

Predicted vs. Gold Tags

Suppose $\mathbf{w}$ was not yet trained well and Viterbi yields:

$$\hat{\mathbf{y}} = (\mathrm{DET}, \mathrm{DET}, \mathrm{VERB}, \mathrm{PRON}, \mathrm{NOUN}).$$

- The first two words are "The" (predicted DET is correct) and "dog" (predicted DET is incorrect!).
- Gold $\mathbf{y}$: (DET, NOUN, VERB, PRON, NOUN).
- The difference is at position 2: $\hat{y}_2 = \mathrm{DET}$ vs. $\mathbf{y}_2 = \mathrm{NOUN}$.

Structured Perceptron Update:

$$\mathbf{w} \leftarrow \mathbf{w} + \Big[\Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{\mathbf{y}})\Big].$$

Feature Difference and Weight Update

Feature Difference $\Phi(\mathbf{x}, \mathbf{y}) - \Phi(\mathbf{x}, \hat{\mathbf{y}})$:

$$\left[\phi_{\text{emit}}(\text{“dog”}, \text{NOUN}) - \phi_{\text{emit}}(\text{“dog”}, \text{DET})\right]$$

$$+ \left[\phi_{\text{trans}}(\text{DET}, \text{NOUN}) - \phi_{\text{trans}}(\text{DET}, \text{DET})\right] + \ldots$$

(Only the parts that differ between the gold and predicted sequences contribute.)

- In our toy representation, $\phi_{\text{emit}}(\text{“dog”}, \text{NOUN}) = 1$ and $\phi_{\text{emit}}(\text{“dog”}, \text{DET}) = 0$.
- So we add weight to the (word=“dog”, tag=NOUN) feature.
- Also fix any transition features that were incorrectly used.

Effect:

$$\mathbf{w}_{(\text{“dog”}, \text{NOUN})} \leftarrow \mathbf{w}_{(\text{“dog”}, \text{NOUN})} + 1, \quad \mathbf{w}_{(\text{DET} \rightarrow \text{DET})} \leftarrow \mathbf{w}_{(\text{DET} \rightarrow \text{DET})} - 1,$$

Result: The correct path is favored next time Viterbi is run.

Summary

- Structured Perceptron $\rightarrow$ extension of the Perceptron to structured outputs.
- In sequence labeling (POS tagging), we use the **Viterbi** algorithm for inference.
- Each update compares the gold sequence with the predicted sequence; weights get "pushed" to favor correct structures.
- Over multiple epochs, the model parameters $\mathbf{w}$ converge to correctly classify training data (if linearly separable in feature space).

Further Reading

Michael Collins.

Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms.

EMNLP, 2002.

Michael Collins.

Parameter Estimation for Statistical Parsing Models: Theory and Practice of Context-Free Grammars.

In *Proceedings of the International Conference on Parsing Technologies*, 2004.

Noah A. Smith.

Linguistic Structure Prediction.

Morgan & Claypool, 2011.