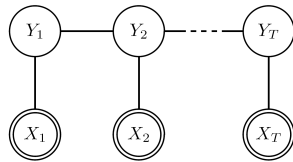


Introduction to Graphical Models

lecture 7 - Perceptron Algorithm for CRFs

Ulf Brefeld
TU Berlin



1/??

Recall: Sequential CRFs

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{i=1}^T \psi^{obs}(X_i, Y_i) \prod_{i=2}^T \psi^{trans}(Y_{i-1}, Y_i)$$

- potential functions:

$$\psi^{trans}(Y_i, Y_{i-1}) = \exp \left\{ \sum_{j=1}^{d_{trans}} w_j^{trans} \phi_j^{trans}(Y_{i-1}, Y_i) \right\}$$

$$\psi^{obs}(X_i, Y_i) = \exp \left\{ \sum_{j=1}^{d_{obs}} w_j^{obs} \phi_j^{obs}(X_i, Y_i) \right\}$$

- $Z(\mathbf{x}) = \sum_{y_1, \dots, y_T} \prod_{i=1}^T \psi^{obs}(X_i, Y_i) \prod_{i=2}^T \psi^{trans}(Y_{i-1}, Y_i)$

2/??

Exemplary Basis Functions

- label-label indicator functions:

$$\phi_1^{trans}(Y_{i-1}, Y_i) = [[Y_{i-1} = \text{noun} \wedge Y_i = \text{noun}]]$$

$$\phi_2^{trans}(Y_{i-1}, Y_i) = [[Y_{i-1} = \text{noun} \wedge Y_i = \text{verb}]]$$

⋮

$$\phi_{d_{trans}}^{trans}(Y_{i-1}, Y_i) = [[Y_{i-1} = \text{adverb} \wedge Y_i = \text{adverb}]]$$

- label-observation indicators:

$$\phi_1^{obs}(X_i, Y_i) = [[X_i = \text{Aachen} \wedge Y_i = \text{noun}]]$$

$$\phi_2^{obs}(X_i, Y_i) = [[X_i = \text{Aar} \wedge Y_i = \text{noun}]]$$

⋮

$$\phi_{d_{obs}}^{obs}(X_i, Y_i) = [[X_i = \text{ZZ-top} \wedge Y_i = \text{adverb}]]$$

3/??

Joint Feature Representation

- Joint representation of input and output variables:

$$\Phi(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^T \phi^o(x_i, y_i)', \sum_{i=2}^T \phi^t(y_{i-1}, y_i)' \right)'$$

- Rewrite conditional probability:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \left\{ \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle \right\}$$

- Observation:

$$P(\mathbf{y}|\mathbf{x}) \propto \langle \mathbf{w}, \Phi(\mathbf{x}, \mathbf{y}) \rangle$$

- MAP estimate:

$$\hat{\mathbf{y}} = \underset{\bar{\mathbf{y}}}{\operatorname{argmax}} P(\bar{\mathbf{y}}|\mathbf{x}) = \operatorname{argmax} \langle \mathbf{w}, \Phi(\mathbf{x}, \bar{\mathbf{y}}) \rangle$$

4/??

Example

- \mathbf{x} = Bob jagt den Hund
- We want

$$[N, V, A, N] = \underset{\bar{y}}{\operatorname{argmax}} \langle \mathbf{w}, \Phi(\mathbf{x}, \bar{y}) \rangle$$

5/??

Example

- \mathbf{x} = Bob jagt den Hund
- We want

$$[N, V, A, N] = \underset{\bar{y}}{\operatorname{argmax}} \langle \mathbf{w}, \Phi(\mathbf{x}, \bar{y}) \rangle$$

- Equivalent representation:

$$\langle \mathbf{w}, \Phi(\mathbf{x}, [N, V, A, N]) \rangle > \langle \mathbf{w}, \Phi(\mathbf{x}, [A, A, A, A]) \rangle$$

$$\langle \mathbf{w}, \Phi(\mathbf{x}, [N, V, A, N]) \rangle > \langle \mathbf{w}, \Phi(\mathbf{x}, [A, A, A, N]) \rangle$$

$$\langle \mathbf{w}, \Phi(\mathbf{x}, [N, V, A, N]) \rangle > \langle \mathbf{w}, \Phi(\mathbf{x}, [A, A, N, A]) \rangle$$

$$\vdots > \vdots$$

$$\langle \mathbf{w}, \Phi(\mathbf{x}, [N, V, A, N]) \rangle > \langle \mathbf{w}, \Phi(\mathbf{x}, [V, V, V, V]) \rangle$$

6/??

Example Contd.

- Another equivalent representation:

$$\langle \mathbf{w}, \Phi(\mathbf{x}, [N, V, A, N]) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}, [A, A, A, A]) \rangle > 0$$

$$\langle \mathbf{w}, \Phi(\mathbf{x}, [N, V, A, N]) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}, [A, A, A, N]) \rangle > 0$$

$$\langle \mathbf{w}, \Phi(\mathbf{x}, [N, V, A, N]) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}, [A, A, N, A]) \rangle > 0$$

$$\vdots \quad \vdots$$

$$\langle \mathbf{w}, \Phi(\mathbf{x}, [N, V, A, N]) \rangle - \langle \mathbf{w}, \Phi(\mathbf{x}, [V, V, V, V]) \rangle > 0$$

- The other way round:
 - Update weight vector \mathbf{w} in case of an error:

$$\langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) \rangle - \max_{\bar{y}} \langle \mathbf{w}, \Phi(\mathbf{x}_i, \bar{y}) \rangle < 0$$

7/??

Primal Perceptron

- Simplify things:
 - Error: $\mathbf{y}_i \neq \hat{\mathbf{y}} = \underset{\bar{y}}{\operatorname{argmax}} \langle \mathbf{w}, \Phi(\mathbf{x}_i, \bar{y}) \rangle$

- Recall gradient of CRF:

$$\frac{\partial \log \mathcal{L}}{\partial \mathbf{w}} = \underbrace{\mathbf{E}_{p(X,Y)}[\Phi(X, Y)]}_{\text{truth/emp. distr.}} - \underbrace{\sum_{i=1}^n \mathbf{E}_{p(Y|\mathbf{x}_i; \mathbf{w})}[\Phi(Y, \mathbf{x}_i)]}_{\text{prediction of model/model distr.}}$$

- Perceptron: perform gradient steps if i -th example is incorrect:

$$\mathbf{w} \leftarrow \mathbf{w} + \underbrace{\Phi(\mathbf{x}_i, \mathbf{y}_i)}_{\text{true pair}} - \underbrace{\Phi(\mathbf{x}_i, \hat{\mathbf{y}})}_{\text{erroneous prediction}}$$

8/??

Primal Perceptron Algorithm

- 1 loop $r = 1, \dots, r_{max}$
- 2 loop $i = 1, \dots, n$
- 3 Compute $\hat{y} = \operatorname{argmax}_{\bar{y}} \langle \mathbf{w}, \Phi(\mathbf{x}_i, \bar{y}) \rangle$
- 4 If $y_i \neq \hat{y}$
- 5 Update $\mathbf{w} \leftarrow \mathbf{w} + \Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, \bar{y})$
- 6 End (if)
- 7 End loop (i)
- 8 End loop (r)

9/??

Convergence

Theorem (Extension of Novikoff)

Given n labeled examples $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$, with $y_i \in \mathcal{Y}(\mathbf{x}_i)$. Let r be the radius of the smallest hypersphere enclosing all difference vectors $\Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, \bar{y})$, for all i and $\bar{y} \neq y_i$,

$$r = \max_{1 \leq i \leq n} \max_{\substack{\bar{y} \in \mathcal{Y}(\mathbf{x}_i) \\ \bar{y} \neq y_i}} |\Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, \bar{y})|.$$

If there exists a vector \mathbf{w}^* such that

$$\forall_{i=1}^n \forall_{\bar{y} \in \mathcal{Y}(\mathbf{x}_i)} \langle \mathbf{w}^*, \Phi(\mathbf{x}_i, y_i) \rangle - \langle \mathbf{w}^*, \Phi(\mathbf{x}_i, \bar{y}) \rangle \geq \bar{\gamma} \quad (1)$$

holds for some $\bar{\gamma} > 0$ then the number of update steps of the generalized perceptron algorithm is upper bounded by

$$\left(\frac{r}{\bar{\gamma}}\right)^2 |\mathbf{w}|^2. \quad (2)$$

10/??

Proof of Theorem

Proof. The weight vector is initialized with $\mathbf{w}^{(0)} = \mathbf{0}$. Let $t > 0$ indicate the t -th error of the generalized perceptron, that is for some $1 \leq i \leq n$

$$y_i \neq \hat{y}_i = \operatorname{argmax}_{\bar{y} \in \mathcal{Y}(\mathbf{x}_i)} \langle \mathbf{w}^{(t-1)}, \Phi(\mathbf{x}_i, \bar{y}) \rangle.$$

The corresponding update step is given by

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} + \Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, \hat{y}_i) \quad (3)$$

Multiplying Equation 3 with the optimal weight vector \mathbf{w}^* yields

$$\begin{aligned} \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle &= \langle \mathbf{w}^*, \mathbf{w}^{(t-1)} \rangle + \langle \mathbf{w}^*, \Phi(\mathbf{x}_i, y_i) \rangle - \langle \mathbf{w}^*, \Phi(\mathbf{x}_i, \hat{y}_i) \rangle \\ &\geq \langle \mathbf{w}^*, \mathbf{w}^{(t-1)} \rangle + \bar{\gamma} \end{aligned}$$

Applying the principle of induction gives us $\langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle \geq t\bar{\gamma}$.

11/??

Proof of Theorem (Contd.)

Now we bound $|\mathbf{w}^{(t)}|^2$ from above by

$$\begin{aligned} |\mathbf{w}^{(t)}|^2 &= \langle \mathbf{w}^{(t-1)} + \Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, \hat{y}_i), \mathbf{w}^{(t-1)} + \Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, \hat{y}_i) \rangle \\ &= |\mathbf{w}^{(t-1)}|^2 + 2\langle \mathbf{w}^{(t-1)}, \Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, \hat{y}_i) \rangle + |\Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, \hat{y}_i)|^2 \\ &\leq |\mathbf{w}^{(t-1)}|^2 + |\Phi(\mathbf{x}_i, y_i) - \Phi(\mathbf{x}_i, \hat{y}_i)|^2 \\ &\leq |\mathbf{w}^{(t-1)}|^2 + r^2. \end{aligned}$$

Thus, by induction we have $|\mathbf{w}^{(t)}|^2 \leq tr^2$. Putting everything together gives us

$$\begin{aligned} t\bar{\gamma} &\leq \langle \mathbf{w}^*, \mathbf{w}^{(t)} \rangle \\ &\leq |\mathbf{w}^*| |\mathbf{w}^{(t)}| \\ &\leq |\mathbf{w}^*| \sqrt{tr}. \end{aligned}$$

Solving for t implies the upper bound

$$t \leq \left(\frac{r}{\bar{\gamma}}\right)^2 |\mathbf{w}^{(*)}|^2.$$

12/??

Towards Dual Perceptrons

- Observation: $\mathbf{w}^{(0)} \leftarrow \mathbf{0}$
- i -th example violates constraint:
 - Update: $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \langle \mathbf{w}, \Phi(\mathbf{x}_i, \mathbf{y}_i) \rangle - \max_{\bar{\mathbf{y}}} \langle \mathbf{w}, \Phi(\mathbf{x}_i, \bar{\mathbf{y}}) \rangle$
- Idea: remember how many times the pair $(\mathbf{x}_i, \bar{\mathbf{y}})$ is used for an update!
 - Variable $\alpha_i(\bar{\mathbf{y}})$ acts as a counter
 - Initialize: $\alpha_i(\bar{\mathbf{y}}) \leftarrow 0$
 - Update: $\alpha_i(\bar{\mathbf{y}}) \leftarrow \alpha_i(\bar{\mathbf{y}}) + 1$
- The α are bound to violated constraints!

13/??

Dual Representation

- Dual parameters
 - $\alpha_i(\bar{\mathbf{y}})$ is proportional to the importance of $\langle \Phi(\mathbf{x}_i, \mathbf{y}_i) \rangle - \Phi(\mathbf{x}_i, \bar{\mathbf{y}})$
- Recall: α counted the number of updates for \mathbf{w}
 - we can thus write:

$$\mathbf{w} = \sum_{i=1}^n \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) (\langle \Phi(\mathbf{x}_i, \mathbf{y}_i) \rangle - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}))$$

- Sparse representation
 - Generally, there are exponentially many $\bar{\mathbf{y}} \neq \mathbf{y}$
 - However, only a few of them will have an $\alpha_i(\bar{\mathbf{y}}) > 0$
 - Feature vector can efficiently be encoded and stored (compare dimensionality of primal and dual!)

14/??

Dual Decision Function

$$\mathbf{w} = \sum_{i=1}^n \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) (\langle \Phi(\mathbf{x}_i, \mathbf{y}_i) \rangle - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}))$$

- Plug dual representation of \mathbf{w} into decision function:

$$\begin{aligned} f(\mathbf{x}', \mathbf{y}') &= \langle \mathbf{w}, \Phi(\mathbf{x}', \mathbf{y}') \rangle \\ &= \left\langle \sum_{i=1}^n \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) (\langle \Phi(\mathbf{x}_i, \mathbf{y}_i) \rangle - \Phi(\mathbf{x}_i, \bar{\mathbf{y}})), \Phi(\mathbf{x}', \mathbf{y}') \right\rangle \\ &= \sum_{i=1}^n \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) (\langle \Phi(\mathbf{x}_i, \mathbf{y}_i) - \Phi(\mathbf{x}_i, \bar{\mathbf{y}}), \Phi(\mathbf{x}', \mathbf{y}') \rangle) \\ &= \sum_{i=1}^n \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) (\langle \Phi(\mathbf{x}_i, \mathbf{y}_i), \Phi(\mathbf{x}', \mathbf{y}') \rangle - \langle \Phi(\mathbf{x}_i, \bar{\mathbf{y}}), \Phi(\mathbf{x}', \mathbf{y}') \rangle) \end{aligned}$$

15/??

Kernels and the Dual Perceptron

- Define $K(\mathbf{x}, \mathbf{y}, \mathbf{x}', \mathbf{y}') = \langle \Phi(\mathbf{x}, \mathbf{y}), \Phi(\mathbf{x}', \mathbf{y}') \rangle$
 - K is called kernel
 - computes inner product in space spanned by Φ
 - rewrite $f(\mathbf{x}, \mathbf{y})$ in terms of kernel functions:

$$f_D(\mathbf{x}', \mathbf{y}') = \sum_{i=1}^n \sum_{\bar{\mathbf{y}} \neq \mathbf{y}_i} \alpha_i(\bar{\mathbf{y}}) (K(\mathbf{x}_i, \mathbf{y}_i, \mathbf{x}', \mathbf{y}') - K(\mathbf{x}_i, \bar{\mathbf{y}}, \mathbf{x}', \mathbf{y}'))$$

- Example (sequences, indicator functions)

$$\begin{aligned} K(\mathbf{x}, \mathbf{y}, \bar{\mathbf{x}}, \bar{\mathbf{y}}) &= \langle \Phi(\mathbf{x}, \mathbf{y}), \Phi(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \rangle \\ &= \sum_{s,t} [[y^{s-1} = \bar{y}^{t-1} \wedge y^s = \bar{y}^t]] \\ &+ \sum_{s,t} [[y^s = \bar{y}^t]] K_x(x^s, \bar{x}^t) \end{aligned}$$

16/??

Kernels on Tokens

- Kernel K_x computes similarity of two tokens
 - Simplest case: $K_x(x, x') = [[x == x']]$
 - No generalization!
- A better choice:
 - K_x computes similarity of feature vectors of observations
 - e.g., n -grams, surface clues
 - Let $\psi(x)$ be the feature vector of token x , then

$$K_x(x, x') = \langle \psi(x), \psi(x') \rangle$$

- K_x can be precomputed for the training process

17/??

Dual Perceptron Algorithm

- 1 loop $r = 1, \dots, r_{max}$
- 2 loop $i = 1, \dots, n$
- 3 Compute $\hat{y} = \operatorname{argmax}_{\bar{y}} f_D(\mathbf{x}_i, \bar{y})$
- 4 If $\mathbf{y}_i \neq \hat{y}$
- 5 Increment $\alpha_i(\hat{y}) \leftarrow \alpha_i(\hat{y}) + 1$
- 6 End (if)
- 7 End loop (i)
- 8 End loop (r)

- Convergence
 - see Collins (2002) and Altun et al. (2003)

18/??

What about the Argmax?

- For dual perceptron it's easy!
- Decompose $f(\mathbf{x}, \mathbf{y}) = f_1(\mathbf{x}, \mathbf{y}) + f_2(\mathbf{x}, \mathbf{y})$ with

$$f_1(\mathbf{x}, \mathbf{y}) = \sum_{\sigma, \tau} a(\sigma, \tau) \sum_s [[y^{s-1} = \sigma \wedge y^s = \tau]]$$

$$a(\sigma, \tau) = \sum_{i, \bar{y} \neq \mathbf{y}_i} \alpha_i(\bar{y}) \sum_t [[\bar{y}^{t-1} = \sigma \wedge \bar{y}^t = \tau]]$$

- and

$$f_2(\mathbf{x}, \mathbf{y}) = \sum_{s, \sigma} [[y^s = \sigma]] \sum_{i, t} b(i, t, \sigma) K_x(x^s, x_i^t),$$

$$b(i, t, \sigma) = \sum_{\mathbf{y} \neq \mathbf{y}_i} [[y^t = \sigma]] \alpha_i(\mathbf{y})$$

- (homework: show that $f = f_1 + f_2$!)

19/??

Correspondence to Viterbi Algorithm

- $a(\sigma, \tau)$ corresponds to transition probabilities $P(y_t = \tau | y_{t-1} = \sigma)$
- for observation scores compute:
 - $B_i^{s\sigma} = \sum_j \sum_t b(j, t, \sigma) k(x_i^s, x_j^t)$
 - $B_i^{s\sigma}$ corresponds to $P(x_{i,s} | y_s = \sigma)$
- Note that a and b (or B) are scores and can be interpreted as log-probs.
- a and B can be directly plugged into log-Viterbi algorithm
- Equivalence between log-Viterbi ($\log(P(\mathbf{y}|\mathbf{x}))$) and $f(\mathbf{x}, \mathbf{y})$

20/??

Named Entity Recognition

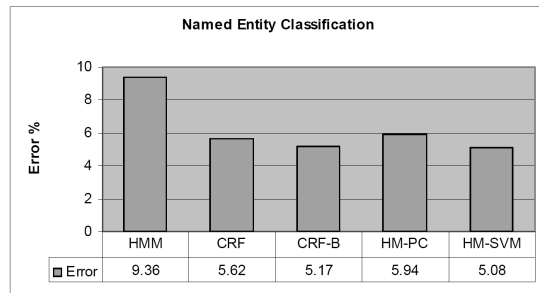


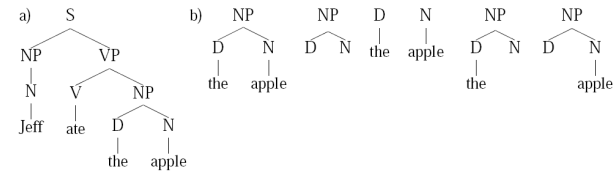
Figure 1. Test error of NER task over a window of size 3 using 5-fold cross validation.

Example: Como (O) contrapartida (O) Deutsche (C-B) Telekom (C-I) vender (O) al (O) consorcio (O) francs (O) su (O) participacion (O) del (O) por (O) ciento (O) en (O) el (O) empresa (O) mixta (O) britnica (O) MetroHoldings (C-B).

(see Altun et al. (2003))

21/??

Natural Language Parsing



Depth	1	2	3	4	5	6
Score	73 ± 1	79 ± 1	80 ± 1	79 ± 1	79 ± 1	78 ± 0.01
Improvement	-1 ± 4	20 ± 6	23 ± 3	21 ± 4	19 ± 4	18 ± 3

Table 1: *Score* shows how the parse score varies with the maximum depth of sub-tree considered by the perceptron. *Improvement* is the relative reduction in error in comparison to the PCFG, which scored 74%. The numbers reported are the mean and standard deviation over the 10 development sets.

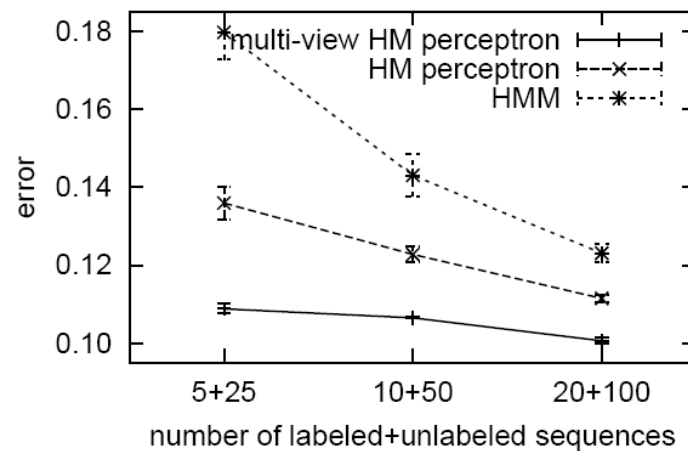
(see Collins&Duffy, 2002)

22/??

BioCreative

- Detection of gene and protein names in biomedical abstracts

BioCreative learning curve



(Brefeld et al., 2005)

23/??

Summary

- Perceptrons for CRFs
 - aka generalized/structured perceptron
- pos:
 - easy to implement
 - efficient training process
- neg:
 - depends on ordering
 - no confidences
 - only 0/1 loss

24/??

Outlook

- Remedy: structural SVMs!