

# 1 Lecture 11 – Structured Prediction

**Remark 1.1.**

$$\min_{\mathbf{w}} \frac{C}{2} \|\mathbf{w}\|_2^2 + \sum_{i \in \mathcal{D}} \epsilon \ln \sum_{\hat{y}} \exp \frac{L(y^{(i)}, \hat{y}) + F(\mathbf{w}, x^{(i)}, \hat{y})}{\epsilon} - F(\mathbf{w}, x^{(i)}, y^{(i)})$$

The general loss function above can be interpolated or specified to be obtain either the multiclass logistic loss function or the multiclass svm loss function.

To obtain the multiclass logistic loss function, set  $\epsilon = 1$  and  $L = 0$ .

$$\begin{aligned} & \sum_{i \in \mathcal{D}} \log \left( \sum_{\hat{y}} \exp (\hat{y} w^T \phi(x) - y_i w^T \phi(x)) \right) \\ &= - \sum_{i \in \mathcal{D}} \log \frac{1}{\left( \sum_{\hat{y}} \exp (\hat{y} w^T \phi(x) - y_i w^T \phi(x)) \right)} \end{aligned}$$

Now do some additional manipulation

$$= - \sum_{i \in \mathcal{D}} \log \frac{\exp (y_i w^T \phi(x))}{\left( \sum_{\hat{y}} \exp \left( \frac{\hat{y} w^T \phi(x)}{1} \right) \right)}$$

If we set  $\epsilon \rightarrow 0$  and  $L = 1$ , then we recover the SVM multiclass classification function.

**Definition 1.2.** A prediction problem is to find  $y^*$  where

$$y^* = \arg \max_{\hat{y}} F(w, x, \hat{y})$$

where  $y^*$  is some vector of values.

**Remark 1.3.** A first temptation is to frame every such maximization problem as a maximization over each of its subcomponents.

That is, maximize

$$\sum_{y_d \in y^*} F_d(w, x, y_d)$$

but doing so would ignore that, oftentimes, there is structural knowledge that aids in the learning problem. That is, oftentimes, we gain information by looking at several components collectively.

**Example 1.4.** If we were given the four letters to the far left and classified them by looking exclusively at each letter, we might think that we should predict Q V I Z. In reality, however, if we look at the letters holistically, we can predict QUIZ.



At one extreme, we can perform a holistic evaluation by solving

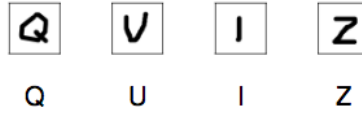
$$\max_{D \in A^4} F_D(w, x, y_D)$$

where  $|A|$  is the set of letters. There are far too many possibilities here. At the other extreme, we can perform discretization by solving

$$\sum_{i=1}^n \max_{d \in A} F_d(w, x_i, y_d)$$

This does not take into account correlations between letters.

**Remark 1.5.** We can perform structural prediction over pairs of attributes, for this reason:



Example:

$$F(w, x, y_1, \dots, y_4) = f_1(w, x, y_1) + f_2(w, x, y_2) + f_3(w, x, y_3) + f_4(w, x, y_4) \\ + f_{1,2}(w, x, y_1, y_2) + f_{2,3}(w, x, y_2, y_3) + f_{3,4}(w, x, y_3, y_4)$$

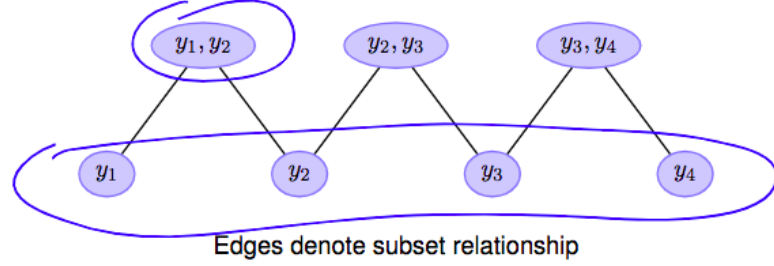
How many function values need to be stored if  $y_d \in \{1, \dots, 26\} \forall d$ ?

$$\text{Earlier: } 26^4 \quad \text{vs. now} \quad 3 \times 26^2 (+4 \cdot 26)$$

This gives us  $3 \times 26^2 + 4(26)$  labelings that we consider. This can be visualized as a diagram:

Visualization of the decomposition:

$$F(\mathbf{w}, x, y_1, \dots, y_4) = f_1(\mathbf{w}, x, y_1) + f_2(\mathbf{w}, x, y_2) + f_3(\mathbf{w}, x, y_3) + f_4(\mathbf{w}, x, y_4) \\ + f_{1,2}(\mathbf{w}, x, y_1, y_2) + f_{2,3}(\mathbf{w}, x, y_2, y_3) + f_{3,4}(\mathbf{w}, x, y_3, y_4)$$



**Remark 1.6.** There are a few algorithms one can employ to perform this search:  
One is Exhaustive Search:

### Exhaustive Search

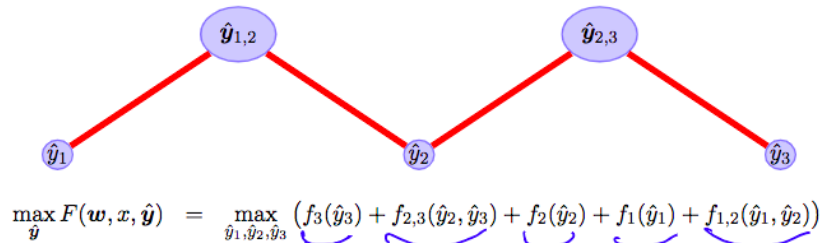
$$\mathbf{y}^* = \arg \max_{\hat{\mathbf{y}}} \sum_r f_r(\mathbf{w}, x, \hat{\mathbf{y}}_r)$$

Algorithm:

- try all possible configurations  $\hat{\mathbf{y}} \in \mathcal{Y}$
- keep highest scoring element
  
- **Advantage:** very simple to implement
- **Disadvantage:** very slow for reasonably sized problems:  $K^D$

Another is dynamic programming:

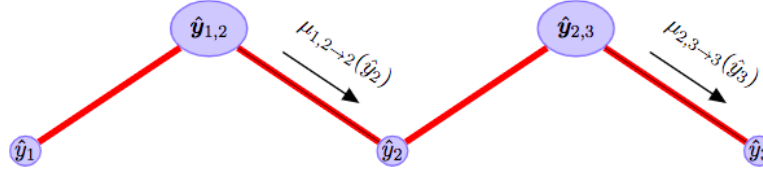
### Dynamic Programming



Recall that a problem admits a solution via dynamic programming if it satisfies two criterion: a problem can be decomposed into subproblems, and these subproblems overlap; a problem has the principle of optimality which is that the optimal solution to a problem is a function of the optimal solution of some of its subproblems.

Above, we see that if we define one problem  $P$  to be maximization of the objective function over all 5 nodes, then this problem has a subproblem  $S$ : maximization over all nodes but  $\hat{y}_3$ . After  $S$  is solved, solving  $P$  then requires maximizing over the nodes involving  $\hat{y}_3$ . Whence, this problem has the principle of optimality and admits a DP solution.

This is formalized by splitting the application of max as below.



$$\begin{aligned}
 \max_{\hat{y}} F(w, x, \hat{y}) &= \max_{\hat{y}_2, \hat{y}_3} (f_3(\hat{y}_3) + f_{2,3}(\hat{y}_2, \hat{y}_3) + f_2(\hat{y}_2) + f_1(\hat{y}_1) + f_{1,2}(\hat{y}_1, \hat{y}_2)) \\
 &= \max_{\hat{y}_3} \left( f_3(\hat{y}_3) + \max_{\hat{y}_2} \left( f_{2,3}(\hat{y}_2, \hat{y}_3) + f_2(\hat{y}_2) + \underbrace{\max_{\hat{y}_1} \{f_1(\hat{y}_1) + f_{1,2}(\hat{y}_1, \hat{y}_2)\}}_{\mu_{1,2 \rightarrow 2}(\hat{y}_2)} \right) \right) \\
 &= \max_{\hat{y}_3} \left( f_3(\hat{y}_3) + \max_{\hat{y}_2} (f_{2,3}(\hat{y}_2, \hat{y}_3) + f_2(\hat{y}_2) + \mu_{1,2 \rightarrow 2}(\hat{y}_2)) \right)
 \end{aligned}$$

*Handwritten notes:* "Not used for y2" with an arrow pointing to the inner max over y2 in the second line.

This is colloquially called “message” passing, because we pass the optimal solution obtained from maximizing  $\hat{y}_1$  and  $\hat{y}_{1,2}$  to a problem maximizing over  $\hat{y}_2$  and then pass that problem to a problem maximizing over  $\hat{y}_3$ .

The summary of DP is as follows. Note that  $K$  is the amount of values that any one dimension of  $y_D$  can take on. Thus, if there is a tree of  $D$  nodes that each involve maximization over a pair of dimensions in  $y_D$ , then there are  $D \cdot K^2$  values that we must work through.

### Dynamic Programming (message passing)

When is this approach suitable?

When the graph is a **tree** (sometimes after re-organizing terms)

- **Advantage:** better complexity than exhaustive search:  $D \cdot K^2$  for pairwise models
- **Disadvantage:** only works for trees

What to do for general loopy graphs?

- Integer Linear Programs
- Linear Programming relaxations
- Dynamic programming extensions (message passing)
- Graph cut algorithms

<++>