University of Illinois Fall 2018

ECE 544NA: Pattern Recognition
Lecture 12: Structured Prediction (Octobert 21)

Lecturer: Alexander Schwing Scribe: Haowen Jiang

1 Purpose

The goal of this lecture is to get to know structured prediction and figure out basic structured inference alogorithms.

2 Recap

The previous lecture explains several classification model such as linear regression, logistic regression, SVM and so on. There is general frameword formular for these score functions.

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

The loss function here including log-loss, hinge-loss and so on. The main purpose when doing classification or regression is find the highset score of the function. The highest express the output we adopt finally.

3 Introduction

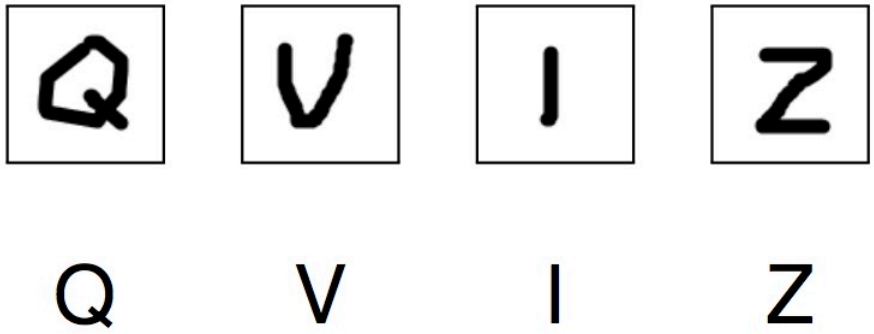
Structured Prediction is a framework to solve classification problem where the data variables are mutually dependent and related. In machine learning, the way of data fitting is to find a function f to map input and output. Sometimes the predictions is a sequence, a sentence or a graph rather than a scalar or a simple classification. In other word, the data hides addition information in the structure. The way to predict output is called structure prediction.

In general, structure prediction will not output a scalar (regression question) or a class(classification problem). It always output a structure which means the output is a tree, a graph or a sequence where elements in it can influence each other.

4 Structured Prediction

4.1 Problem and Example

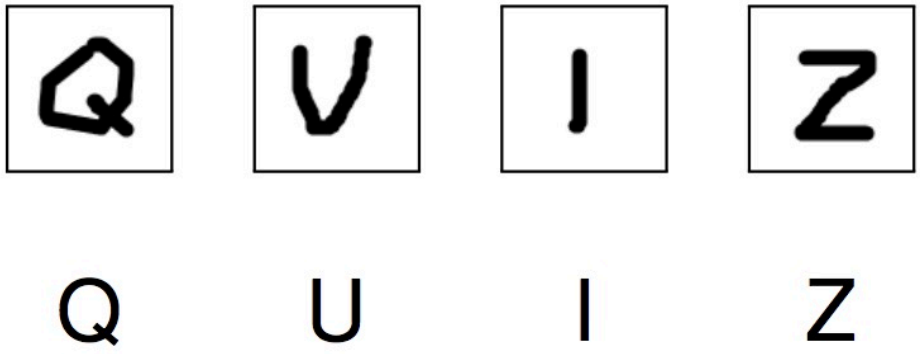
One of the common and easy method to solve structure classification problem is to predict each section and combine the whole output together. Assume now four figures of characters are offered and the sequences compose to a word: QUIZ. The independent classification of each letter is Q, V, I, Z separately.



4.2 Solution

The way to take correlation into account is formulate the whole input into a entirety. As all we know, there are 26 capital letters exist in alphabet. Therefore, for every figure output from the example above, we can get 26 scores with different values. Since the question has four letters to predict, the size of output space is 26^4 which requires a quite large output space. Imagine now a sentence or a graph requires to recognize. Assume the sentence has 100 words in and each word has 5 letters, the output space will be 26^{500} which is a problem.

The structure prediction output where correlations taken into account shows below.



4.3 Application

For the graph, structure prediction can smooth the graph and denoise since much more independent false prediction can be rectify. For example, when doing image binarization or geting the grayscale value, stucture prediction can help to get a smooth output rather than a noisy one. Assume the formal prediction of certain pixel is white where all the surrounding pixels are black, the structured prediction can rectify this pixel into black. In that case, much more false positive pixels could be rectify which smooth the graph in some degree.



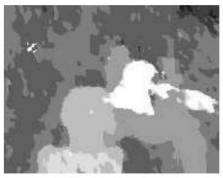


Image Independent Prediction **Structured Prediction**

Similarity, the structure prediction model works for noise reduction. Most of the noise can be reducted by analysis the surrounding pixels.

A simple graph can still be solved without correlation. However, the accuray will decrease since much structure information is ignored by this way. In fact, structure prediction always used to estimate a complex object, including image segmentation, sentence parsing, protein folding, stereo vision and so on. For the image segmentation, we can treat it as a label classification problem. Sentence parseing questions means the model should parse a tree. Protein folding is to parse a protein structure and stereo vision is to estimate a disparity map.

Image segmentation

From the graph shows above, the model need to classificate which pixels belong to human, bike and background seperately.

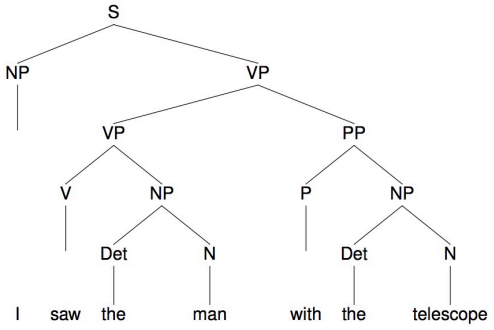
Sentence parsing

The whole sentense can be split into several words. The assignment is to recoginize each word and combine together. However, the correlations between words can help to keep sentense composition right.

$$x^{(i)} \rightarrow y^{(i)} = (y_1^{(i)}, \ldots, y_D^{(i)})$$

$$x^{(i)} \rightarrow y^{(i)} = (y_1^{(i)}, \ldots, y_D^{(i)})$$

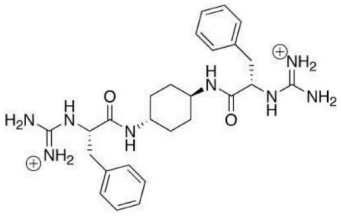
I saw the man with the telescope.

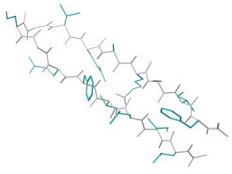


Protein folding

Each protein structure will influence the other protein.

$$x^{(i)} \rightarrow y^{(i)} = (y_1^{(i)}, \ldots, y_D^{(i)})$$

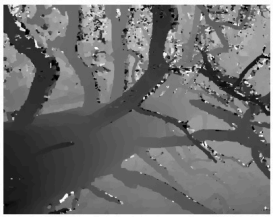




Stereo vision

Nearby pixels could offer more information which will help to rectify prediction value.

$$x^{(i)} \quad \rightarrow \quad \boldsymbol{y}^{(i)} = (y_1^{(i)}, \ldots, y_D^{(i)})$$



5 Exhautive Search

5.1 Problem of Exhautive Search

Assume the model has D output to predict and the range of each output is K. In that case, the amount of possibilities to store and explore is K^D. Too large space and too many calculations needed.

$$\max_{\hat{\boldsymbol{y}}} F(\boldsymbol{w}, x, \hat{y}_1, \ldots, \hat{y}_D) = \max_{\hat{\boldsymbol{y}}} \sum_{d=1}^{D} f_d(\boldsymbol{w}, x, \hat{y}_d) = \sum_{d=1}^{D} \max_{\hat{y}_d} f_d(\boldsymbol{w}, x, \hat{y}_d)$$

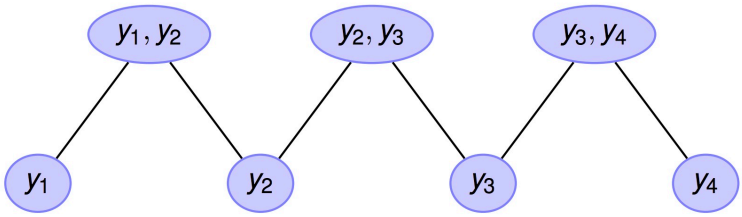
If we give up to predict the whole variables but predict every variable in space K, the correlation information will be missed.

5.2 Example and Solution

The way to solve exhaustive search problem is to explore the nearby information to do the prediction. In other words, set a restriction of the whole set to reduce the store space and calculations. Back to the problem of word quiz prediction, we can set the restrictions to 2 which means we just need to get score value of{1}, {2}, {3}, {4}, {1, 2}, {2, 3} and {3, 4}(numbers here mean the index of the letters). Since the {1, 2} contains {1} and {2} information in so that {1} and {2} can be omitted.

$$\begin{aligned} F(\boldsymbol{w}, x, y_1, \ldots, y_4) = & f_1(\boldsymbol{w}, x, y_1) + f_2(\boldsymbol{w}, x, y_2) + f_3(\boldsymbol{w}, x, y_3) + f_4(\boldsymbol{w}, x, y_4) \\ & + f_{1,2}(\boldsymbol{w}, x, y_1, y_2) + f_{2,3}(\boldsymbol{w}, x, y_2, y_3) + f_{3,4}(\boldsymbol{w}, x, y_3, y_4) \end{aligned}$$

Retrieve to the exhaustive search, the amount of possibilities to be store or explored is 26^4. Consider every letter as the node in the graph, the edges will denote the subset relationship or colleration. THe amount of nodes possibility is $4 * 26$. Meanwhile, the amount possibilities of edges is $3 * 26^2$. Thus, the current output space is $3 * 26^2 (+4 * 26)$.



6 Markov/Conditional random field

6.1 Problem

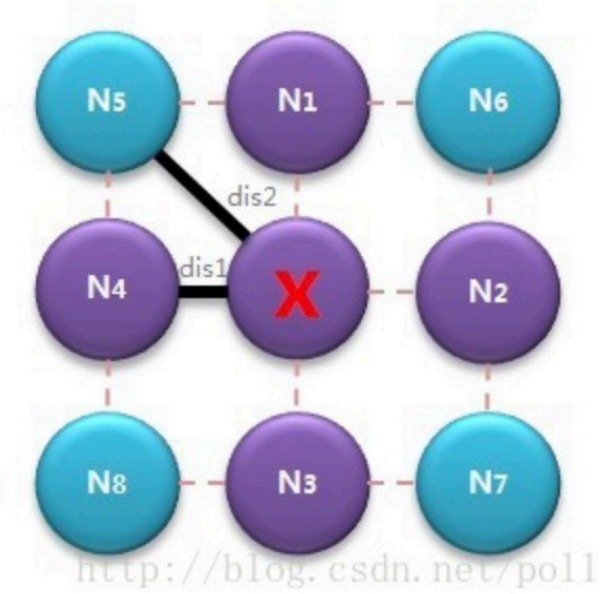
The above discussion is a special case where only unary variables get into considered. Some other questions such as stereo vision is multi-variables predicted rather than unary. One of the models to solve this problem is the Markov/Conditional random field.

6.2 Difinition

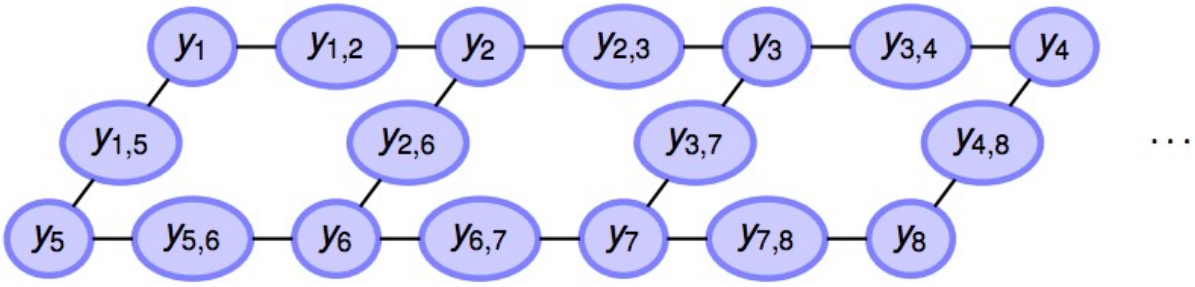
Markov/Conditional random progress is one of the random progress which is extended from Markov Chain. The main feature is: When the current condition is known, the future transformation will only depend on current condition rather than past. For example, the weather of today will only related to yesterdays weather. The day before yesterdays information will not take into account.

Random field is that every position of a phase space place a value with certain random distribution. Assume that we plant crops into an area. Random field means that we need to decided which field planted which kind of crops.

Markov random field is one of the special random progress. The value of the variable is multi-vector even space dot rather than real number value. For example, when filling letters into a chessboard, the letter in every grid will only correlate to the nearby grid. When dealing with graph problem, one pixel's feature is much easier to be influenced by neaby pixels. The influence from surrounding pixels will decrease with the distance increase.



From the aboce graph, X is any pixel in the graph. The purple pixel is the neighborhood of X where the distance between them is 1. The blue dots are the pixels where the distance is $\sqrt{2}$. In real, pixel X will be chosen as the center, surrounding pixels with certain distance as the neighbor pixels. The distance in this example can be calculated by Euclidean distance. The blue pixels has less influence campared to the purple pixels.



From the figure shows above, the subscibes with only one number is that simple classification. The subscibes with more numbers is the correlation variables. Thus, the score function can be formulated:

$$F(\boldsymbol{w}, x, \boldsymbol{y}) = \sum_{d=1}^{D} f_d(\boldsymbol{w}, x, y_d) + \sum_{i,j} f_{i,j}(\boldsymbol{w}, x, y_i, y_j)$$

The final score is composed by the sum of every node's score and the sum of edge's score.

The unary term can solve the image evidence questions. The pairwise term can use to image smothness.

7 Inference and Algorithms

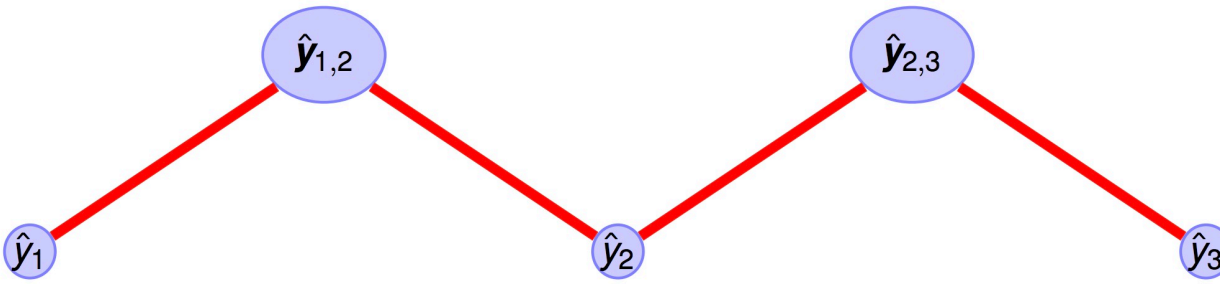
Several algorithms can be adpoted to find the highest scoring configuration. From the previous lectures, we can get: Exhaustive search, Dynamic programming, Integer linear program, Linear programming relaxation, Message passing and Graph-cut.

7.1 Exhaustive Search

$$\boldsymbol{y}^* = \arg\max_{\hat{\boldsymbol{y}}} \sum_{r} f_r(\boldsymbol{w}, x, \hat{\boldsymbol{y}}_r)$$

Exhaustive search should explore all possible configuretions and iterate the whole score to find the highest score element. In fact, the exhaustive search is quite simple to implement but required a large output space and a complex calculation when the dataset is large. Thus, it is extremely slow to solve a reasonable size problem. From the discussion above, we can get the complexicity of the problem is K^D.

7.2 Dynamic Programming



For the dynamic programming, the whole problem will be sovle into sections. Assume we get the graph shows above, the first step is to get highest score of node(y_1) and edge($y_{1,2}$). Then calculating node(y_2) and edge($y_{2,3}$) and so on. Treat the sum score of node(y_1) and edge($y_{1,2}$) as $\mu_{1,2} \rightarrow 2(y_2)$. Keep going it until to the last. Thus, we can get:

$$\max_{\hat{\boldsymbol{y}}} F(\boldsymbol{w}, x, \hat{\boldsymbol{y}}) = \max_{\hat{y}_1, \hat{y}_2, \hat{y}_3} \left(f_3(\hat{y}_3) + f_{2,3}(\hat{y}_2, \hat{y}_3) + f_2(\hat{y}_2) + f_1(\hat{y}_1) + f_{1,2}(\hat{y}_1, \hat{y}_2) \right)$$

$$= \max_{\hat{y}_3} \left(f_3(\hat{y}_3) + \max_{\hat{y}_2} \left(f_{2,3}(\hat{y}_2, y_3) + f_2(\hat{y}_2) + \underbrace{\max_{\hat{y}_1} \left\{ f_1(\hat{y}_1) + f_{1,2}(\hat{y}_1, \hat{y}_2) \right\}}_{\mu_{1,2 \rightarrow 2}(\hat{y}_2)} \right) \right)$$

$$= \max_{\hat{y}_3} \left(f_3(\hat{y}_3) + \max_{\hat{y}_2} \left(f_{2,3}(\hat{y}_2, \hat{y}_3) + f_2(\hat{y}_2) + \mu_{1,2 \rightarrow 2}(\hat{y}_2) \right) \right)$$

Obviously, the highest score could be solved by step by step until the connection ended. However, the limitation of the dynamic programming is that the dynamic programming can only be used in a tree graph. Whe the graph is connected end to end(loopy graphs), this algorithms will be trapped into a circle and get into trouble.

Actually, dynamic programming is better configured than exhaustive search. From the discussion above, the complexcity of the dynamic programming is $D * K^2$.

8 Loopy Graphs

Since dynamic programming only works for tree graph, some other models should be used to sove loopy graphs. There are several algorithms to solve loopy graphs but will not go deep in this lecture. The algorithms including: Integer Linear Programs, Linear Programming relaxations, Dynamic programming extensions (message passing) and Graph cut algorithms.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.