Name:

University of Illinois Spring 2020

CS 446/ECE 449 Machine Learning
Homework 6: Structured Prediction

Due on Thursday March 12 2020, noon Central Time

1. **[28 points]** Structured Prediction

We are interested in jointly predicting/modeling two discrete random variables $y = (y_1, y_2) \in \mathcal{Y}$ with $y_i \in \mathcal{Y}_i = \{0, 1\}$ for $i \in \{1, 2\}$ and $\mathcal{Y} = \prod_{i\in\{1,2\}} \mathcal{Y}_i$. We define the joint probability distribution to be $p(y) = p(y_1, y_2) = \frac{1}{Z} \exp F(y)$.

(a) (3 points) What is the value of Z (in terms of $F(y)$) and what is Z called? How many configurations do we need to sum over? Provide the expression using $\mathcal{Y}_i$.

Your answer:

$$P = \frac{\cancel{\exp}(F(y))}{\sum_{y\in\mathcal{Y}} \exp(F(y))} = \frac{1}{Z}\cancel{\exp F(y)}$$

$$Z = \sum_{y\in\mathcal{Y}} \exp(F(y))$$

$$\#\text{ configs} = \left|\prod_{i\in\{1,2\}} \mathcal{Y}_i\right|$$

4 configs for this problem

Z is GIBBS Measure, essentially normalizing our scores to use as probabilities.

(b) (6 points) Next we want to solve (for any hyperparameter ϵ)

$$\max_{\hat{p}\in\Delta_{\mathcal{Y}}} \sum_{y\in\mathcal{Y}} \hat{p}(y)F(y) - \sum_{y\in\mathcal{Y}} \epsilon\hat{p}(y)\log\hat{p}(y), \tag{1}$$

where $\Delta_{\mathcal{Y}}$ denotes the probability simplex, *i.e.*, $\hat{p}$ is a valid probability distribution over its domain $\mathcal{Y}$. Using general notation, write down the Lagrangian and compute its derivative w.r.t. $\hat{p}(y)$ $\forall y \in \mathcal{Y}$. Subsequently, find the optimal $\hat{p}^*$. What is the resulting optimal cost function value for the program given in Eq. (1)? How does this result relate to part (a)?

Your answer:

$$L(\cdot) = -\sum_{y\in\mathcal{Y}} \hat{p}(y)F(y) + \sum_{y\in\mathcal{Y}} \epsilon\hat{p}(y)\log\hat{p}(y) + \sum_{y\in\mathcal{Y}} \lambda(\hat{p}(y) - 1)$$

$$\frac{\partial L}{\partial \hat{p}(y)} \Rightarrow -\sum_{y\in\mathcal{Y}} F(y) - \sum_{y\in\mathcal{Y}} \epsilon(1+\log(\hat{p}(y))) + \sum_{y\in\mathcal{Y}}\lambda = 0$$

$$\prod_{y\in\mathcal{Y}} \hat{p}(y) = \prod_{y\in\mathcal{Y}} \underbrace{\exp\left(\frac{F(y)-\lambda}{\epsilon} - 1\right)}_{\text{each } \hat{p}(y)}$$

$$\hat{p}^*(y) = \frac{\exp(\frac{1}{\epsilon}(F(y)-\lambda)-\epsilon)}{\sum_{y\in\mathcal{Y}} \exp(\frac{1}{\epsilon}(F(y)-\lambda)-\epsilon)}$$

$$\hat{p}^*(y) = \frac{\exp\left(\frac{F(y)}{\epsilon}\right)}{\sum_{y\in\mathcal{Y}} \exp\left(\frac{F(y)}{\epsilon}\right)}.$$

$$\sum \frac{\exp(\frac{F(y)}{\epsilon})F(y)}{\sum\exp(\frac{F(y)}{\epsilon})} - \sum \frac{\exp(\frac{F(y)}{\epsilon})\cdot F(y) - \epsilon\exp(\frac{F(y)}{\epsilon})\log(\sum\exp\frac{F(y)}{\epsilon})}{\sum\exp(\frac{F(y)}{\epsilon})}.$$

$$\sum \frac{\epsilon\exp(F(y)/\epsilon)\log(\sum\exp\frac{F(y)}{\epsilon})}{\sum\exp(\frac{F(y)}{\epsilon})}.$$

$$\frac{\epsilon\log\sum\exp(\frac{F(y)}{\epsilon})}{\cancel{\sum\exp(\frac{F(y)}{\epsilon})}}\cdot\cancel{\sum\exp\left(\frac{F(y)}{\epsilon}\right)}$$

$$\boxed{\text{cost}(\hat{p}^*(y)) = \epsilon\log\sum\exp\left(\frac{F(y)}{\epsilon}\right).}$$

This is the log of Z we found in part a. additionally, we now have ϵ which acts as temperature allowing us to switch between different models and gives rise to the generic multiclass framework we see in class. For ex, when $\epsilon = 1$ we have logistic regression and when $\epsilon \to 0$ we have SVM.

Name: ____________________

(c) (3 points) For the program in Eq. (1) assume now $\epsilon = 0$, *i.e.*, we are searching for that configuration $y^* = \arg\max_{\hat{y}\in\mathcal{Y}} F(\hat{y})$ which maximizes $F(y)$. Assume $F(y) = f_1(y_1) + f_2(y_2) + f_{1,2}(y_1, y_2)$. How many different values can the functions f_1, f_2 and $f_{1,2}$ result in?

Your answer:

$f_1 = |\{0,1\}| = 2$

$f_2 = |\{0,1\}| = 2$

$f_{12} = |\{(0,0),(0,1),(1,0),(1,1)\}| = 4$

8 different values.

(d) (9 points) As discussed in class, finding the global maximizer can be equivalently written as the following integer linear program:

$$\max_b \sum_{r,y_r} b_r(y_r) f_r(y_r) \qquad \text{s.t.} \qquad \begin{cases} b_r(y_r) \in \{0,1\} & \forall r, y_r \\ \sum_{y_r} b_r(y_r) = 1 & \forall r \\ \sum_{y_p \setminus y_r} b_p(y_p) = b_r(y_r) & \forall r, p \in P(r), y_r \end{cases}. \tag{2}$$

Using the decomposition $F(y) = f_1(y_1)+f_2(y_2)+f_{1,2}(y_1,y_2)$, *i.e.*, for $r \in \{\{1\},\{2\},\{1,2\}\}$, explicitly state the integer linear program and all its constraints for the special case that $\mathcal{Y}_i = \{0,1\}$ for $i \in \{1,2\}$. (**Hint:** The parent sets are as follows: $P(\{1\}) = \{1,2\}$ and $P(\{2\}) = \{1,2\}$. Use notation such as $f_1(y_1 = 0)$ and $b_1(y_1 = 0)$.)

Your answer:

$$\max_{b_1,b_2,b_{12}} \begin{bmatrix} b_1(y_1=0) \\ b_1(y_1=1) \\ b_2(y_2=0) \\ b_2(y_2=1) \\ b_{12}(y_1=0,y_2=0) \\ b_{12}(y_1=0,y_2=1) \\ b_{12}(y_1=1,y_2=0) \\ b_{12}(y_1=1,y_2=1) \end{bmatrix}^T \begin{bmatrix} f_1(y_1=0) \\ f_1(y_1=1) \\ f_2(y_2=0) \\ f_2(y_2=1) \\ f_{12}(y_1=0,y_2=0) \\ f_{12}(y_1=0,y_2=1) \\ f_{12}(y_1=1,y_2=0) \\ f_{12}(y_1=1,y_2=0) \end{bmatrix}$$

s.t.

$b_1(y_1=0) \in \{0,1\}$ $\quad$ $b_2(y_2=0) \in \{0,1\}$

$b_1(y_1=1) \in \{0,1\}$ $\quad$ $b_2(y_2=1) \in \{0,1\}$

$b_1(y_1=0) + b_1(y_1=1) = 1$

$b_2(y_2=0) + b_2(y_2=1) = 1$

$b_{12}(y_1=0,y_2=0) + b_{12}(y_1=0,y_2=1) + b_{12}(y_1=0,y_2=1) + b_{12}(y_1=1,y_2=1) = 1$

$b_{12}(y_1=0,y_2=0) + b_{12}(y_1=0,y_2=1) = b_1(y_1=0)$

$b_{12}(y_1=1,y_2=0) + b_{12}(y_1=1,y_2=1) = b_1(y_1=1)$

$b_{12}(y_1=0,y_2=0) + b_{12}(y_1=1,y_2=0) = b_2(y_2=0)$

$b_{12}(y_1=0,y_2=1) + b_{12}(y_1=1,y_2=1) = b_2(y_2=1)$

Name: ____________________

(e) (3 points) Let b be the vector

$$
\begin{aligned}
b = [\quad & b_1(y_1=0), b_1(y_1=1), b_2(y_2=0), b_2(y_2=1), \\
& b_{1,2}(y_1=0,y_2=0), b_{1,2}(y_1=1,y_2=0), b_{1,2}(y_1=0,y_2=1), b_{1,2}(y_1=1,y_2=1)]^\top .
\end{aligned}
$$

Specify all but the integrality constraints of part (d) using matrix vector notation, *i.e.*, provide A and c for $Ab = c$.

Your answer:

$$
A = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
-1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & -1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 1 & 1
\end{bmatrix}
\qquad
c = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

(f) (4 points) Complete `A6_Structure.py` where we approximately solve the integer linear program using the linear programming relaxation. Implement the constraints. Why do we provide $-f$ as input to the solver? What is the obtained result b for the relaxation of the program given in Eq. (2) and its cost function value? Is this the configuration y^* which has the largest score?

Your answer:

By default, the solver minimizes the problem. We want to maximize so we provide it with -f.

This is not the configuration with the highest score. From the potentials in the code, we see the highest score should 7 as we would the configuration would be:

$b_1(y_1=1)=1 \Rightarrow 1$

$b_2(y_2=0)=1 \Rightarrow 1$

$b_{12}(y_1=1, y_2=0)=1 \Rightarrow 5$

$1+1+5=7$

$$
b = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}
\qquad
\text{cost} = -5
$$