Statistical structured prediction

Question set (Part II)

Victoria Beltrán Domínguez

February 2022

Contents

1 Question 1 (2 points)

Briefly explain the differences between Classification and Structured Output Prediction.
Cite two application examples each paradigm.

The difference between classification and structured-output prediction is clear. Whereas a classification problem tries to classify a sample into a category given a finite set of categories (normally small), structured-output prediction aims to predict an structued sequence, graph, set, etc.

Note that classification problems have a bounded set of possible labels whereas structued-output prediction deals with a possibly infinite space.

An example of classification problems is detecting if a person has lung cancer or not based on medical data. Another example could be sentiment analysis: identifying an emotion from a given text or document.

On the other hand, one example of structured-output prediction problem could be text synthesis (given a text, generate some kind of summary) or image description (given an image, provide a brief description of the things that can be seen).

2 Question 2 (2 points)

Justify why the naive Bayes decomposition of Eq.(5) is adequate for karyotype recognition problem.

Naive Bayes decomposition used in Eq.(5) simplifies and reduces de computational cost of the algorithm by assuming that the hypothesis made for each chromosome only depends on the chromosome itself and not on the rest of the chromosomes. It seems to adecuate perfectly to the problem we are trying to solve, simplifying the computation of the likelihood.

3 Question 3 (2 points)

Briefly explain all the steps and assumptions needed to derive Eq.(9) from Eq.(7).

In this exercise, our goal will be to transform Eq.(7) into Eq.9 step by step. For that, let us remember Eq.(7):

$$\hat{y} = argmax_{h \in \mathcal{H}} P(h|x, h', f)$$

The first step is to transform the feedback into a feedback decoding function $d : \mathcal{F} \rightarrow \mathcal{D}$, that maps each raw feedback signal f into its corresponding (trivial and unique) decoding ($d = d(f)$). Therefore, feedback f can be replaced by its decoding d.

$$\hat{y} = argmax_{h \in \mathcal{H}} P(h|x, h', d)$$

The next step would be to apply Bayes' rule (removing those elements that do not depend on h, as they are not important in an argmax):

$$\hat{y} = argmax_{h \in \mathcal{H}} \frac{P(h, x, h', d)}{P(x, h', d)} = argmax_{h \in \mathcal{H}} P(h, x, h', d)$$

Using the chain rule, we can transform the previous equation in:

$$\hat{y} = argmax_{h \in \mathcal{H}} P(h')P(d|h')P(h|h', d)P(x|h', d, h)$$

Removing all those elements that do not depend on h we have:

$$\hat{y} = argmax_{h \in \mathcal{H}} P(h|h', d)P(x|h', d, h)$$

Finally, taking into account that $P(x|h', d, h)$ is independent of h' and d given h (as the new hypothesis already contains the changes induced by the feedback from the previous hypothesis), we get Eq.(9):

$$\hat{y} = argmax_{h \in \mathcal{H}} P(h|h', d)P(x|h)$$

4 Question 4 (4 points)

In the Interactive Karyotyping problem, assume we have, for each test karyotype, a likelihood matrix P[] where $P[i, c] = P(x_i|c) 1 \leq i \leq 22$ $1 \leq c \leq 22$ (the individual chromosome likelihoods considered in page 16 and the in the product of Eq.(5)). Write a C (or pseudo-code) algorithm that implements a greedy solution to the interactive (pasive, left-to-right) search problem discussed in pages 23-27 (see the basic idea in page 18).

The first thing to notice in this exercise is that since we are assuming that the classes are equiprobable, $max_c P(c|x_j) = max_c P(x_j|c)$, and thus, we can directly compute the maximum and the hypothesis from the given data.

In order to develop this exercise, I will be writting pseudocode (as I do not feel confident writting in C). Before starting, i will define two previous funcions:

- **sorted_indexes(probabilities)**: This function will receive an array containing the maximum likelihood probability for each sample, and will return an ordered array of sample indexes. For example, imagine we execute the function as sorted([0.8,0.9,0.3]). This means that for sample 0, maximum probability is 0.8, for sample 1 is 0.9 and for sample 2 is 0.3. The function will return an array [1,0,2] denoting the order of the samples we will have to visit. Note that each position of the returning array will be denoting the index of a sample.
- **pred(idx_sample, K)**: This function will receive the index of the sample to look and a set of possible hypothesis left, and from all the hypothesis left, will return the one with the maximum likelihood probability for that sample.

Once we have defined the previous function, our greedy algorithm is defined as:

```
Input: set of samples x, likelihood matrix P
Output: Sequence of classified karotypes
res = []
maxprob = []
for(i = 0;i<len(x);i++){
    maxprob[i] = max(P[i,:])
}
s = sorted_indexes(maxprob)
K = {"1","2","3",...,"22"}
for(i = 0;i<len(x);i++){
    n_sample = s[i]
    hyp = pred(n_sample,K)
    res.append((n_sample, hyp))
    K = K - hyp
}
return res
```

5 Question 6 (3 points)

Briefly explain all the steps and assumptions needed to derive Eq.(19) from Eq.(7).

In this exercise, our goal will be to transform Eq.(7) into Eq.(19) step by step. For that, let us remember Eq.(7):

$$\hat{h} = argmax_{h \in \mathcal{H}} P(h|x, h', f)$$

We can transform the previous ecuation by adding the decodification marginalization in such way that:

$$\hat{h} = argmax_{h \in \mathcal{H}} \sum_{d} P(h, d|x, h', f)$$

Once we have transform it in terms of decodification, we apply Bayes rule (removing those elements whose value does not depend on h):

$$\hat{h} = argmax_{h \in \mathcal{H}} \frac{\sum_d P(h, d, x, h', f)}{P(x, h', f)} = argmax_{h \in \mathcal{H}} \sum_{d} P(h, d, x, h', f)$$

We can apply the mode approximation to the previously obtained equation, such as:

$$\hat{h} \approx argmax_{h \in \mathcal{H}} max_d P(h, d, x, h', f)$$

Once we have a joint probability, we break down the probabilities using the chain rule:

$$\hat{h} \approx argmax_{h \in \mathcal{H}} max_d P(h')P(d|h')P(f|h', d)P(h|h', d, f)P(x|h', d, f, h)$$

Finally, by ruling out constant values and finding independencies, we get that:

- $P(h')$ can be ruled out because it is a constant term.
- $P(f|h', d)$ is independent of h' given d, so we can simplify the expression as $P(f|d)$. The independence comes from the fact that the feedback does already take into account the previous history.
- $P(h|h', d, f)$ is independent of f given h' and d, so we can simplify the expression as $P(h|h', d)$. That is because the decoding information has all the needed information of the feedback.
- $P(x|h', d, f, h)$ is independent of h', d and f given h, so we can simplify the expression as $P(x|h)$, as the new hypothesis has already taken into account the history, the decodification and its feedback.

Applying all the previously mentioned simplifications, we can obtain the expression corresponding to that from Eq.(19):

$$\hat{h} \approx argmax_{h \in \mathcal{H}} max_d P(f|d)P(d|h')P(x|h)P(h|h', d)$$

6 Question 8 (2 points)

Briefly explain the concepts and main differences between Active and Passive interaction protocols.

In a passive interaction protocol, the system launches the hypothesis and the user corrects the samples. Basically, the user itself is responsible for choosing which samples the system has labeled correctly and which ones are mislabeled. As we can see, the user has to correct (either explicitly or implicitly) all samples, so all samples are supervised by the user.

In an active interaction protocol, it is the system that determines the hypotheses to be corrected by the user, generally sending the user the samples whose probability is below a certain confidence. However, in the active interaction protocol, the user has to correct the sample determined by the system. This method aims to facilitate the task to the user, sacrificing precision.