IE531: Algorithms for Data Analytics
End-Term Exam
Spring, 2020

1. (*10 points*) (**Gibbs Sampling**) Consider the 4-state Markov Chain shown in figure 1 with desired stationary distribution

$$\pi_{(1,1)} = \pi_{(2,2)} = a \text{ and } \pi_{(1,2)} = \pi_{(2,1)} = b$$

   (a) (*5 points*) Using the Gibbs Sampling Procedure covered in class compute the 4×4 probability matrix $\mathbf{P}$

$$\mathbf{P} = \begin{pmatrix} p_{(1,1),(1,1)} & p_{(1,1),(1,2)} & p_{(1,1),(2,1)} & p_{(1,1),(2,2)} \\ p_{(1,2),(1,1)} & p_{(1,2),(1,2)} & p_{(1,2),(2,1)} & p_{(1,2),(2,2)} \\ p_{(2,1),(1,1)} & p_{(2,1),(1,2)} & p_{(2,1),(2,1)} & p_{(2,1),(2,2)} \\ p_{(2,2),(1,1)} & p_{(2,2),(1,2)} & p_{(2,2),(2,1)} & p_{(2,2),(2,2)} \end{pmatrix}$$

   with the desired stationary distribution.
   Using the Gibbs Sampling Procedure we get

$$\mathbf{P} = \begin{pmatrix} \frac{a-b}{a+b} & \frac{b}{a+b} & \frac{b}{a+b} & 0 \\ \frac{a}{a+b} & \frac{b-a}{a+b} & 0 & \frac{a}{a+b} \\ \frac{a}{a+b} & 0 & \frac{b-a}{a+b} & \frac{a}{a+b} \\ 0 & \frac{b}{a+b} & \frac{b}{a+b} & \frac{a-b}{a+b} \end{pmatrix}$$

   (b) (*5 points*) Verify your answer to problem 1a.

$$\begin{pmatrix} a & b & b & a \end{pmatrix} \begin{pmatrix} \frac{a-b}{a+b} & \frac{b}{a+b} & \frac{b}{a+b} & 0 \\ \frac{a}{a+b} & \frac{b-a}{a+b} & 0 & \frac{a}{a+b} \\ \frac{a}{a+b} & 0 & \frac{b-a}{a+b} & \frac{a}{a+b} \\ 0 & \frac{b}{a+b} & \frac{b}{a+b} & \frac{a-b}{a+b} \end{pmatrix} = \begin{pmatrix} a & b & b & a \end{pmatrix}$$

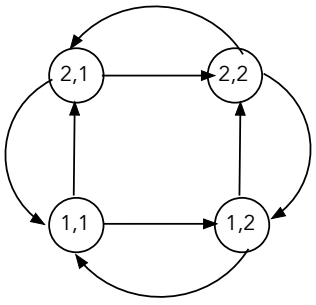


Figure 1: The Markov Chain for Problem 1.

2. (*20 points*) (**Linear Separability**) We have training data $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in \mathcal{R}^d$, and let us suppose each $\mathbf{x}_i$ has a label $y_i \in \{-1, +1\}$. The set of data points $\mathbf{x}_i$ with $y_i = +1$ (resp. $y_i = -1$) will be referred to as the set of *positive* (resp. *negative*) examples, and will be denoted by $\mathcal{P}$ (resp. $\mathcal{N}$).

   In class we looked at Pedroso and Murata's LP that can be used to check if the data is linearly separable/non-separable. We suppose there is a hyperplane $H_{\mathbf{w},b} : \mathbf{w}^T\mathbf{x} + b = 0$ that separates the positive from negative examples. For non-separable data we require

   (a) $\forall \mathbf{x}_i \in \mathcal{P}, \mathbf{w}^T\mathbf{x}_1 + b \geq +1 - \xi_i$,

   (b) $\forall \mathbf{x}_i \in \mathcal{N}, \mathbf{w}^T\mathbf{x}_1 + b \leq -1 + \xi_i$,

   (c) $\xi_i \geq 0, \forall i$, and

   with the objective function

$$\underset{\mathbf{w}, b, \xi}{\text{minimize}} \quad \left\{ \|w\|^2 + \left(\sum_i \xi_i \right) \right\}.$$

   Following some observations involving *conjugate-norms*, this resulted in the following LP

$$\begin{array}{ll} \text{minimize} & a + C\left(\sum_{i\in\mathcal{P}} \xi_i^+ + \sum_{i\in\mathcal{N}} \xi_i^-\right) \\ & \mathbf{w}^T\mathbf{x}_i + b \geq 1 - \xi_i^+, \forall i \in \mathcal{P}, \\ & \mathbf{w}^T\mathbf{x}_i + b \leq -1 + \xi_i^-, \forall i \in \mathcal{N}, \\ \text{subject to} & a \geq \mathbf{w}_j, \forall j \in \{1, 2, \ldots, d\}, \\ & a \geq -\mathbf{w}_j, \forall j \in \{1, 2, \ldots, d\}, \\ & a, b \in \mathcal{R}; \mathbf{w} \in \mathcal{R}^d; \forall i, \xi_i^+ > 0; \forall i, \xi_i^- > 0 \end{array}$$

   Use the above LP formulation to investigate the Linear Separability of the following 2-class data sets.

   (a) (*10 points*) Determine if the following 2-class data set is linearly separable, where

$$\mathcal{P} = \{(0,1)^T, (1,0)^T\}$$
$$\mathcal{N} = \{(0,0)^T, (1,1)^T\}$$

   We know that the data set is separable if and only if the ξ_i's are all zero. Assuming $\mathbf{w}^T = (w_1\ w_2)$, while $\xi_1 = \xi_2 = \xi_3 = \xi_4 = 0$, we have the following constraints for Pedroso and Murata's LP

$$w_1 + b \geq 1$$
$$w_2 + b \geq 1$$
$$b \leq -1$$
$$w_1 + w_2 + b \leq -1$$
all variables are non-negative

Note, $(b \leq -1) \Rightarrow (-b \geq 1)$, we will use the inequality $(-b \geq 1)$ in place of the third inequality in the list above. Adding the three top inequalities we get

$$w_1 + w_2 + b \geq 3,$$

while the last inequality in the above list requires

$$w_1 + w_2 + b \leq -1,$$

which means we cannot find a suitable value for w_1, w_2 and b (i.e. the LP is not feasible). Therefore, the data set is not linearly separable.

(b) (*10 points*) Determine if the following 2-class data set is linearly separable, where

$$\mathcal{P} = \{(-1,-1)^T, (0,0)^T, (1,1)^T\}$$
$$\mathcal{N} = \{(-1,1)^T, (1,-1)^T\}$$

Using same logic as the previous problem – the feasible region for the LP is

$$\begin{gathered} -w_1 - w_2 + b \geq 1 \\ b \geq 1 \\ w_1 + w_2 + b \geq 1 \\ -w_1 + w_2 + b \leq -1 \\ w_1 - w_2 + b \leq -1 \end{gathered}$$

Adding the last two inequalities we get $2b \leq -2 \Rightarrow b \leq -1$. The second constraint requires $b \geq 1$. Therefore, is LP is not feasible, and the data set is not linearly separable.

(**PS**: Any attempt as showing these (rather obvious) problems by any other method will not get you any points! To paraphrase Obi-Wan Kenobi – "*Use the LP, Luke!*")

3. (*10 points*) (**Linear Separability using "Kernelization"**) Consider the 2-class data sets of problems 2a and 2b. Obviously, one or more of these data-sets are not linearly separable. Let us suppose the four (resp. five) data points for problem 2a (resp. 2b) are represented as

$$\left\{ \begin{pmatrix} x_i \\ y_i \end{pmatrix} \middle| i \in \{1, \ldots 4\} \text{ for problem 2a or, } i \in \{1, \ldots 5\} \text{ for problem 2b} \right\}$$

where $i \in \{1, \ldots, 4\}$ (resp. $i \in \{1, \ldots, 5\}$) for problem 2a (resp. problem 2b). Using an intuitive explanation (i.e. no need for an LP formulation) show that you can convert the 2-dimensional, non-separable data set into a 3-dimensional, linearly separable data set by using "kernalization." That is, find a function $f(x_i, y_i)$ such that the resulting 3-dimensional data sets

$$\left\{ \begin{pmatrix} x_i \\ y_i \\ f(x_i, y_i) \end{pmatrix} \middle| i \in \{1, \ldots 4\} \text{ for problem 2a or, } i \in \{1, \ldots 5\} \text{ for problem 2b} \right\}$$

are linearly separable (and can be learned, as a consequence).

We know that both data-sets are not linearly separable from the solutions to the previous problem. There are many ways of adding an extra-dimension to these data sets to get at a linearly separable 3-dimensional data set. Here is one possible solution for the data set of problem 2a –

$$f(x_i, y_i) = Cos^2\left(\frac{(x+y)\pi}{2}\right),$$

which results in the data-set

$$\mathcal{P} = \{(0,1,0)^T, (1,0,0)^T\}$$
$$\mathcal{N} = \{(0,0,1)^T, (1,1,1)^T\}$$

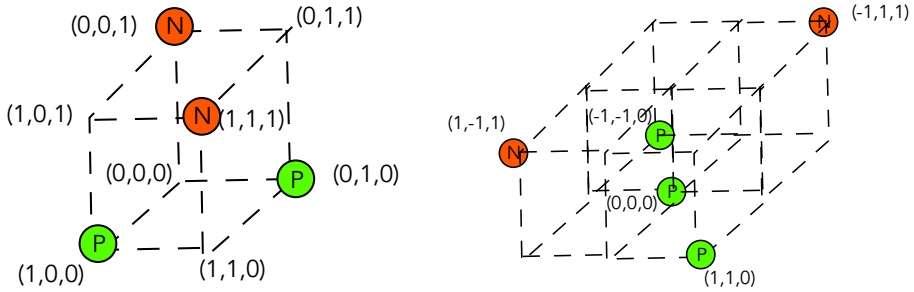
which is linearly separable (cf. visual inspection of figure 2(a)). For problem 2b try –

$$f(x_i, y_i) = |x| \times Cos\left(\frac{(x+y)\pi}{4}\right),$$

which results in the data-set

$$\mathcal{P} = \{(-1,-1,0)^T, (0,0,0)^T, (1,1,0)^T\}$$
$$\mathcal{N} = \{(-1,1,1)^T, (1,-1,1)^T\}$$

which is linearly separable (cf. visual inspection of figure 2(b)).



(a) Problem 2a (b) Problem 2b

Figure 2: The "kernalization" of the non-separable data-sets of problems 2a and 2b to linearly separable data-sets in higher dimensions.

4. (*10 points*) (**Approximate Solution to the** FREQUENT **Problem**) We have a stream $\langle a_1, a_2, \ldots, a_n \rangle$ where each $a_i \in \{1, 2, \ldots m\}$. We have the associated frequency vector $(f_1, f_2, \ldots f_m)$, where $\sum_{i=1}^{n} f_i = n$. We wish to maintain an approximation $(\widehat{f_1}, \widehat{f_2}, \ldots \widehat{f_m})$ to the frequency vector $(f_1, f_2, \ldots f_m)$, such that $\forall j, |f_j - \widehat{f_j}| \leq \epsilon n$, for a desired $\epsilon > 0$.

In section 1.2 of my notes on the material in Chapter 6 of the text, I introduced you a solution to the above problem that kept track of l-many keys (and their associated counters) where $l \ll n$. This procedure required us to maintain at least one empty key (with its counter zeroed-out) at all instants. This was accomplished by subtracting the median of the counter-values from each counter.

We noted in class that it was an overkill to subtract the median from all counter-values, and we could have used the minimum in place of the median. Derive an expression for the value of l (i.e. the number of keys-with-their-counters) that would accomplish the objective when we use the minimum counter-value instead of the median.

I have worked-out the case when

$$\delta_t = \begin{array}{c}\text{median} \\ \text{over } l\text{-many counters}\end{array} \widehat{f_j}$$

This time around we are using

$$\delta_t = \begin{array}{c}\text{minimum} \\ \text{over } l\text{-many counters}\end{array} \widehat{f_j}$$

but the analysis/proof is the practically identical, save the third item in the list shown below (which is slightly-different from the case when we subtracted the median)

(a) $\forall j \in \{1, 2, \ldots, m\}, \widehat{f_j} \leq f_j$,

(b) $\widehat{f_j} \geq f_j - \sum_{t=1}^{n} \delta_t \Rightarrow \sum_{t=1}^{n} \delta_t \geq f_j - \widehat{f_j}$, and

(c) $0 \leq \sum_{j=1}^{m} \widehat{f_j} \leq \sum_{t=1}^{n}(1 - ((l-1) \times \delta_t)) = n - (l-1)\sum_{t=1}^{n} \delta_t \Rightarrow \sum_{t=1}^{n} \delta_t \leq \frac{n}{l-1}$.

Plugging the final-conclusion from item 3 shown above into item 2, we have $\frac{n}{l-1} \geq \sum_{t=1}^{n} \delta_t \geq f_j - \widehat{f_j}$, which means if $l = \frac{1+\epsilon}{\epsilon}, n\epsilon \geq f_j - \widehat{f_j}$. From item 1, we can say $n\epsilon \geq |f_j - \widehat{f_j}|$ (or, if you want to exactly confirm with the requirement of the problem: $|f_j - \widehat{f_j}| \leq n\epsilon$).

Note, that when δ_t was the median, we needed $l = 2/\epsilon$ counters to meet the specification. On the other hand, when δ_t is the minimum value, we needed $l \approx 1/\epsilon$ (i.e. fewer) counters to meet the specification.

5. (*10 points*) (**Deletions in Bloom Filters**) Suppose we built an n-long *Bloom Filter* for m-many objects, using k-many hash-functions. We observed that for regular/vanilla Bloom Filters, there can be no False Negative Errors, while it possible to have False Positive Errors. We showed in class that the

$$Prob\{\text{False Positive Error}\} = (1 - e^{-km/n})^k.$$

and if we set $k = \frac{n}{m} \ln 2 \approx (0.7n)/m$, then

$$Prob\{\text{False Positive Error}\} = (1 - e^{-km/n})^k = (1/2)^k = (1/2)^{(0.7n/m)} = (0.615)^{n/m}.$$

If $Prob\{\text{False Positive Error}\} = p$, then[1]

$$p = (0.615)^{n/m} \Rightarrow \frac{n}{m} = \frac{\log_2 p}{\log_2 0.615} = 0.7 \log_2 p.$$

(a) (*2 points*) Suppose we used a regular/vanilla Bloom Filter, and we naively deleted the k-many bit positions for any object that we thought was in our inventory, but could not find it at the spot is should be. Can we have False Negative Errors after this operation? Explain.

This is sometimes called the "deletion of the false-positive" error. That is, maybe the reason for why the object we thought we should carry was not actually in our inventory was because it was a false-positive error. Zeroing out the k-many bit-locations in the Bloom filter after this will result in false negative errors (something that cannot happen in Bloom Filters under normal circumstances). See taped Lecture for additional details.

(b) (*4 points*) Explain how a *Counting Bloom Filter* is different from regular/-vanilla Bloom Filters. What is the process by which objects get inserted in this filter? What is the process by which objects get deleted from this filter?

To handle the deletion problem the Counting Bloom Filter is has n-many counters (instead of the basic/vanilla Bloom Filter that has n-many bits).

(c) (*4 points*) Without getting into the nitty-gritty mathematical formulae, tell me (in words) how the *number of bits per counter* for a Counting Bloom Filter is decided. What was the logic behind saying 4-bits are sufficient per counter for a typical Counting Bloom Filter.

The punch-line is this – we use about 4-bits per counter for the Counting Bloom Filter (because the probability of $k(= \frac{n}{m} \ln 2 \approx (0.7n)/m)$-many hash-functions sending more than 16 objects to the same spot on the filter is very very small ($< 1.37 \times 10^{-15}$; see the Fan-Almeida-Border paper on Compass or my lecture notes for details).

6. (*20 points*) (**Random Sampling of an Unknown Stream**) Describe a process by which you can choose a single-sample that is uniformly-random from a stream $\langle a_1, a_2, \ldots, a_n \rangle$, when you have no ia priori idea of what n is going to be. Or stated differently, you will only know what n is after you have seen the last member of the stream.

(**PS**: I am looking for (1) a sampling-algorithm, and (2) a proof that if so far we have seen $\langle a_1, a_2, \ldots, a_k \rangle$ where $k \leq n$, then the algorithm would have picked any member of the stream with equal probability (i.e with probability $\frac{1}{k}$.)

This is essentially a case of $s = 1$ in the procedure in section 6 of my notes on the material in Chapter 6 of the text. As soon as a_1 (i.e. the first-stream-member) arrives, you must sample it. The probability of sampling of a k-long stream, where $k = 1$ is $\frac{1}{k} = 1$. This establishes the base-case of induction. Assume so far

[1] As an illustration, we took the case of $p \leq \frac{1}{1000}$ then $\frac{n}{m} \approx 0.7 \log_2 (1/1000) \approx 7$ and $k = 0.7 \times 7 \approx 5$. That is, if we are happy with a 1/1000 chance of a false positive error, then $k \approx 5$ and $n/m \approx 7$, which is quite good.

we saw $\langle a_1, a_2, \ldots, a_k \rangle$, and our currently sampled-value has been picked from these k-many elements with probability $\frac{1}{k}$ (Induction hypothesis). Now, assume a_{k+1} arrives. With probability $\frac{1}{k+1}$ you must pick a_{k+1}; and if it is picked, you have to drop the previous sample. We show that we have effectively sampled the $(k+1)$-long stream with probability $\frac{1}{k+1}$.

For the induction-step, you consider two cases (1) a_{k+1} was picked, and (2) a_{k+1} was not picked. For case 1, we picked the sampled-item (i.e. a_{k+1}) it with probability $\frac{1}{k+1}$ by design (and we are good-to-go). For case 2, we picked the sampled-item from the k-long stream with probability $\frac{1}{k}$ (induction hypothesis). But, now this probability has to be multiplied by the probability that we did not pick a_{k+1} (which is why we continued to hold the previous sample after the $(k+1)$-th item arrived, this will result in an effective probability of selection for case 2 as

$$\underbrace{\left(1 - \frac{1}{k+1}\right)}_{\substack{\text{Prob. that } a_{k+1} \\ \text{was not picked}}} \times \overbrace{\frac{1}{k}}^{\substack{\text{Prob. of sampling before} \\ \text{the } (k+1)\text{-th item arrived}}} = \frac{1}{k+1},$$

which establishes the induction-step for both cases.