CS 344 (Spring 2018): Class Test 3

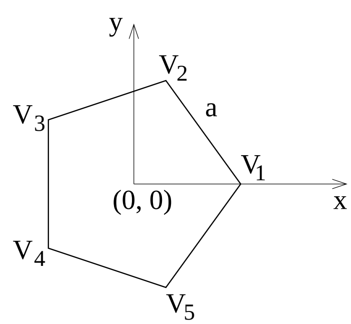
Instructor: Shivaram Kalyanakrishnan

11.05 a.m. – 12.00 p.m., March 28, 2018, 103 New CSE Building

Total marks: 15

Note. Provide brief justifications and/or calculations along with each answer to illustrate how you arrived at the answer.

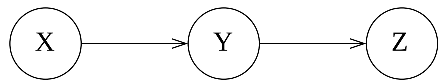
Question 1. Consider a *regular polygon* P_n with $n \geq 3$ sides, each side of length $a > 0$. Assume the polygon is centred at the origin, and one of its vertices lies on the positive x axis. This vertex is named V_1. The remaining vertices, visited in anticlockwise sequence, are named $V_2, V_3, \ldots, V_n$. The figure below shows P_5, a regular pentagon.



Given n, your task is to sample a point (x_0, y_0) uniformly at random from within P_n. In other words, for every two regions R and R' that are contained in P_n, such that R and R' have an equal area, the probability of returning a point from R must be equal to the probability of returning a point from R'. Your only access to randomness is through an operator $random()$, which returns a number drawn uniformly at random from the interval $[0, 1]$. Specify a procedure that uses this operator in order to sample (x_0, y_0) uniformly at random from P_n.

You will get 3 marks if your procedure is correct. Contingent on correctness, a maximum of 2 additional marks will be awarded based on the number of times your procedure invokes $random()$. You should try to minimise this number. If your procedure makes at most 2 calls to $random()$, you will receive the full 2 marks. If the number of calls exceeds 2, but is still upper-bounded by some constant (independent of n and a), you will receive 1 mark. Otherwise, you will receive no marks in addition to the 3 for correctness. [5 marks]

Question 2. Let X, Y, and Z be Boolean random variables. X takes values x and $\neg x$; Y takes values y and $\neg y$; and Z takes values z and $\neg z$. The joint probability distribution of these random variables is given by the Bayes Net shown below; $a, b, c, d, e \in (0, 1)$.



X	$P(X)$
x	a
$\neg x$	1 - a

X	Y	$P(Y\|X)$
x	y	b
x	$\neg y$	$1 - b$
$\neg x$	y	c
$\neg x$	$\neg y$	$1 - c$

Y	Z	$P(Z\|Y)$
y	z	d
y	$\neg z$	$1 - d$
$\neg y$	z	e
$\neg y$	$\neg z$	$1 - e$

2a. If we draw a sample s from the joint distribution—by first sampling X, then Y conditioned on X, and then Z conditioned on Y, what is the probability that $s = (x, \neg y, \neg z)$? [1 mark]

2b. Suppose we draw the sample s as in 2a, but discard it if either $X = \neg x$ or $Z = \neg z$. What is the probability that s will be discarded? [2 marks]

2c. We draw samples from the Bayes Net and discard them as per the rule in 2b. Suppose the first two samples that do not get discarded are $s_1 = (x_1, y_1, z_1)$ and $s_2 = (x_2, y_2, z_2)$. What is the probability that $y_1 = y_2$? [2 marks]

Question 3. Consider the Perceptron Learning Algorithm as we discussed in class, but with the single change that the initial weight vector $\mathbf{w}_0$ is *not* set to the zero vector. Rather, we take it to be an arbitrary vector of finite length. Suppose, as assumed in our class discussion, that (1) the positive and negative points are separable by a hyperplane passing through the origin, and (2) the points are all of a finite length (at most $R \in \mathbb{R}$). Is it guaranteed that this version of the Perceptron Learning Algorithm (with $\mathbf{w}_0 \neq \mathbf{0}$) will converge? Prove that your answer is correct.

If you claim convergence, your proof only needs to show convergence—unlike the proof in class, you need not derive an explicit bound on the number of iterations.

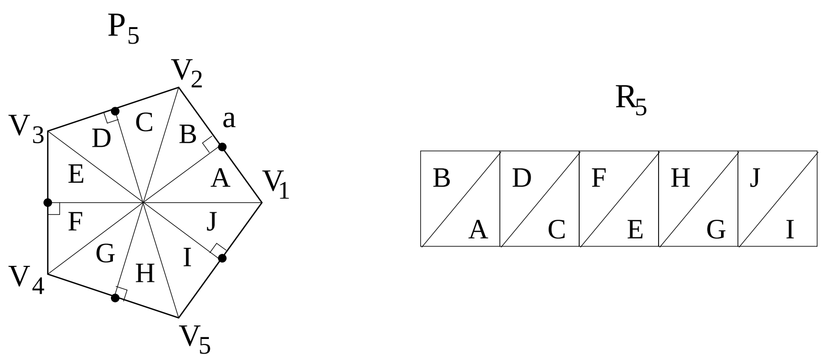
If you claim non-convergence, it would suffice to show a numerical counterexample: a set of points, choice of $\mathbf{w}_0$, and a rule for picking incorrectly classified points, such that the Perceptron will classify at least one point incorrectly at every stage.

Any mathematical derivations you might undertake can include terms such as $\mathbf{w}_0$ and $||\mathbf{w}_0||$. You can also use γ and $\mathbf{w}^\star$ with the same definitions they had in the class discussion. [5 marks]

Solutions

1. One natural approach is to draw a rectangle R that encloses P_n, and to apply rejection sampling: that is, to (1) sample a point uniformly at random from R, (2) return the point if it also lies in P_n, and if not (3) repeat the process. While this method guarantees uniform sampling from P_n, the number of samples it will need from R can be arbitrarily large.

A contrasting approach, using the geometry of regular polygons, allows us to make do with two calls to $random()$. Being a regular polygon, P_n can be partitioned into n triangular "slices" centred at the origin and having as vertices V_i and $V_{(i+1) \bmod n}$, for $i \in \{1, 2, \ldots n\}$. Indeed each of these (congruent) slices is an isosceles triangle, with the sides connecting to the origin being the same length. If we drop a perpendicular from the origin to divide each slice into two right triangles, we obtain a total of $2n$ right triangles: each with one side $a/2$, and the opposing angle π/n. The figure below shows on the left a partitioning of P_5 into 10 such triangles A through J.



The $2n$ triangles obtained above can each be shifted, rotated, and (possibly) mirrored so that they tile a rectangle R_n (shown in the figure on the right for $n = 5$). One side of this rectangle is $\frac{na}{2}$, and the other is $a \tan^{-1}(\frac{\pi}{n})$. We are already familiar with the simple procedure to sample a point (x_R, y_R) uniformly at random from R_n with two calls to $random()$. Since the rectangle is obtained by piecing together portions of the original polygon (with no stretching involved), we can transform (x_R, y_R) to a point (x_P, y_P) inside P_n by undertaking the inverse mirroring, rotation, and translation operations performed to the triangle containing it. This way, (x_R, y_R) will be obtained uniformly at random from P_n.

While the strategy of transforming the polygon into a rectangle is an intuitive way to proceed with this question, it is by no means the only way to sample uniformly from P_n with only two calls to $random()$.

2a. $P(x)P(\neg y|x)P(\neg z|\neg y) = a(1-b)(1-e)$.

2b. The sample s will *not* be discarded only if it is (x, y, z) or $(x, \neg y, z)$. Hence the probability of not discarding the sample is $abd + a(1-b)e$. The probability of discarding the sample is $1 - abd - a(1-b)e$.

2c. The only samples that do not get discarded are (x, y, z) and $(x, \neg y, z)$. The probability that a sample is (x, y, z), given it did not get discarded, is

$$\frac{P(x, y, z)}{P(x, z)} = \frac{abd}{P(x, z)}.$$

The probability that a sample is $(x, \neg y, z)$, given it did not get discarded, is

$$\frac{P(x, \neg y, z)}{P(x, z)} = \frac{a(1-b)e}{P(x, z)}.$$

Normalising, the first probability is $p = \frac{abd}{abd + a(1-b)e}$, and the second probability is $q = \frac{a(1-b)e}{abd + a(1-b)e}$. The required answer is $p^2 + q^2$.

3. If $\mathbf{w}_0 \neq \mathbf{0}$, the Perceptron Learning Algorithm will still converge. If we follow the same line of analysis as we did in class[1], we obtain

$$||\mathbf{w}^{k+1}|| \geq \mathbf{w}_0 \cdot \mathbf{w}^\star + (k+1)\gamma \text{ and } ||\mathbf{w}^{k+1}||^2 \leq ||\mathbf{w}_0||^2 + (k+1)R^2,$$

which imply

$$(\mathbf{w}_0 \cdot \mathbf{w}^\star + (k+1)\gamma)^2 \leq ||\mathbf{w}_0||^2 + (k+1)R^2.$$

The LHS is a *quadratic* function of k, which is increasing in $[\max\{0, -1 - \frac{\mathbf{w}_0 \cdot \mathbf{w}^\star}{\gamma}\}, \infty)$. The RHS is a *linear* function of k, which is increasing in $[0, \infty)$. Clearly the inequality can be satisfied only up to a finite value of k.

[1]Note that our class discussion took the initial weight vector to be $\mathbf{w}_1$; here we take it to be $\mathbf{w}_0$.