ML from Data – HW 5

319238119, 207576463

1. $K(x,y) = (x \cdot y + 1)^3 = (x_1y_1 + x_2y_2 + 1)^3 =$
   $\big((x_1y_1 + x_2y_2 + 1)(x_1y_1 + x_2y_2 + 1)\big)(x_1y_1 + x_2y_2 + 1) =$
   $(x_1^2y_1^2 + x_1x_2y_1y_2 + x_1y_1 + x_1x_2y_1y_2 + x_2^2y_2^2 + x_2y_2 + x_1y_1 + x_2y_2 + 1)(x_1y_1 + x_2y_2 + 1) =$
   $(x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 + 2x_1y_1 + 2x_2y_2 + 1)(x_1y_1 + x_2y_2 + 1) =$
   $x_1^3y_1^3 + x_1^2x_2y_1^2y_2 + x_1^2y_1^2 + 2x_1^2x_2y_1^2y_2 + 2x_1x_2^2y_1y_2^2 + 2x_1x_2y_1y_2 + x_1x_2^2y_1y_2^2 + x_2^3y_2^3 + x_2^2y_2^2 +$
   $2x_1^2y_1^2 + 2x_1x_2y_1y_2 + 2x_1y_1 + 2x_1x_2y_1y_2 + 2x_2^2y_2^2 + 2x_2y_2 + x_1y_1 + x_2y_2 + 1 =$
   $x_1^3y_1^3 + 3x_1^2x_2y_1^2y_2 + 3x_1x_2^2y_1y_2^2 + 6x_1x_2y_1y_2 + 3x_2^2y_2^2 + 3x_1^2y_1^2 + 3x_1y_1 + 3x_2y_2 + x_2^3y_2^3 + 1 =$
   $x_1^3y_1^3 + x_2^3y_2^3 + 3x_1^2x_2y_1^2y_2 + 3x_1x_2^2y_1y_2^2 + 6x_1x_2y_1y_2 + 3x_2^2y_2^2 + 3x_1^2y_1^2 + 3x_1y_1 + 3x_2y_2 + 1$

   a. $\psi(x) = \left(x_1^3, x_2^3, \sqrt{3}x_1^2x_2, \sqrt{3}x_1x_2^2, \sqrt{6}x_1x_2, \sqrt{3}x_2^2, \sqrt{3}x_1^2, \sqrt{3}x_1, \sqrt{3}x_2, 1\right)$
   b. Full rational variety
   c. We saved 10 multiplication operations by using $K(x,y)$ instead $\psi(x)\cdot\psi(y)$

2. $f(x,y) = 2x - y$, $g(x,y) = \frac{x^2}{4} + y^2 = 1 \Rightarrow g(x,y) = \frac{x^2}{4} + y^2 - 1 = 0$

   $$\mathcal{L}(x,y,\lambda) = 2x - y + \lambda\left(\frac{x^2}{4} + y^2 - 1\right)$$

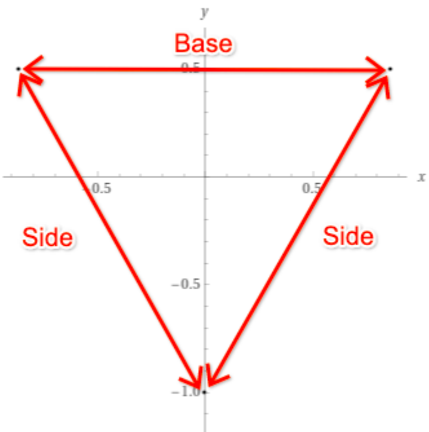
   1. $\frac{\partial}{\partial x}\mathcal{L}(x,y,\lambda) = 2 + \frac{2\lambda x}{4}$
   2. $\frac{\partial}{\partial y}\mathcal{L}(x,y,\lambda) = -1 + 2\lambda y$
   3. $\frac{\partial}{\partial \lambda}\mathcal{L}(x,y,\lambda) = \frac{x^2}{4} + y^2 - 1$

   From 1: $2 + \frac{2\lambda x}{4} = 0 \Rightarrow 2\lambda x = -8 \Rightarrow x = \frac{-4}{\lambda}$
   From 2: $-1 + 2\lambda y = 0 \Rightarrow 2\lambda y = 1 \Rightarrow y = \frac{1}{2\lambda}$
   From 1+2+3: $\frac{\left(\frac{-4}{\lambda}\right)^2}{4} + \left(\frac{1}{2\lambda}\right)^2 - 1 = 0 \Rightarrow \frac{16}{4\lambda^2} + \frac{1}{4\lambda^2} - 1 = \frac{17}{4\lambda^2} - 1 = 0 \Rightarrow \frac{17}{4\lambda^2} = 1 \Rightarrow \frac{17}{4} = \lambda^2 \Rightarrow \lambda_1, \lambda_2 = \pm\sqrt{\frac{17}{4}}$

   $$x_1 = \frac{-4}{\sqrt{\frac{17}{4}}} = -\frac{8\sqrt{17}}{17} \quad y_1 = \frac{1}{\sqrt{\frac{17}{4}}\cdot 2} = \frac{\sqrt{17}}{17} \qquad x_2 = \frac{-4}{-\sqrt{\frac{17}{4}}} = \frac{8\sqrt{17}}{17}, \quad y_2 = \frac{1}{-\sqrt{\frac{17}{4}}\cdot 2} = -\frac{\sqrt{17}}{17}$$

   $$f(x_1,y_1) = 2\cdot -\frac{8\sqrt{17}}{17} - \frac{\sqrt{17}}{17} = -\sqrt{17}$$
   $$f(x_2,y_2) = 2\cdot\frac{8\sqrt{17}}{17} + \frac{\sqrt{17}}{17} = \sqrt{17}$$
   $$(x_1,y_1)\ is\ the\ minimun, (x_2,y_2)\ is\ the\ maximum\ of\ the\ function\ with\ the\ given\ constraint$$

3. Given this piazza post, In this solution we assume the triangle h is of the following form, even though this triangle is not upright as stated in the question. This detail is not critical for the solution complexity but changes some of the details in it.



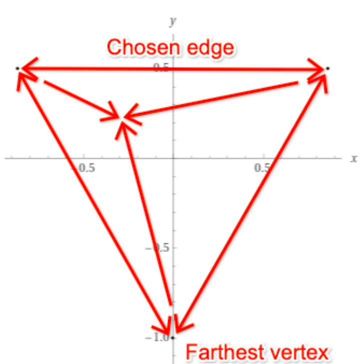
Algorithm to calculate an origin-centered equilateral triangle h given $D = \{(x_i, y_i)\}|_{i=1}^{m}$ a dataset of points of size m in $\mathbb{R}^2$ and C as defined in the question:

1) For every point (x, y) in D:
   a) Calculate the origin-centered equilateral triangle h which contains (x, y) on its perimeter.
   b) Return h if all the other points are contained in it.

We returned the smallest triangle h that contains all the points and hence, given that the data is in fact correct, $h \subseteq C$.

There are two points left to explain:

1) How to implement step 3.1)a) – Given a point (x, y), how to calculate an origin-centered equilateral triangle that contains (x, y) on its perimeter?
   We start by finding which edge of the triangle should contain the point. We do this by finding the farthest triangle vertex from the point, which indicates that the point should be on the edge that is in front of this vertex, as demonstrated in the following diagram.



   At this point we know which edge contains the point (x, y). We also can calculate easily the slopes values of all the triangle edges, because the triangle is origin-centered and equilateral. Using simple math skills we can get the all the 3 linear lines of the form $ax + bx + c = 0$ that form the triangle in a fixed amount of steps denoted as t_1 (we don't write the full logic because it's long, has edge cases and is not the primary purpose of this assignment).

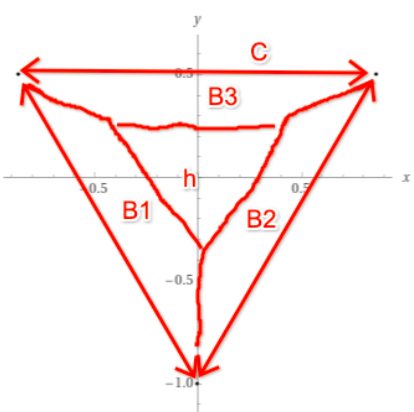
2) How to implement step 3.1)b) - Given a triangle defined by 3 linear lines of the form $ax + bx + c = 0$ of which one is the triangle base and the others are its sides and a point (x, y), how do we check whether the triangle contains (x, y)?
We simply check that (x, y) is **above** the triangle sides and **below** the triangle base. We do that by putting the x value of the point in the line formula and comparing the values of y. This is a simple check that requires a small fixed amount of actions denoted by t_2.

Building a triangle for a point in the set costs t_1 and checking whether all the other points are within it costs mt_2. We do these actions for all the points and therefore the time complexity of this algorithm is $m^2t_2 + mt_1$ which is polynomial on the size of the input data set m.

We now would like to calculate a bound on the number of samples m that ensures with probably δ that $err(h, C) \le \epsilon$. The bound will be of the form $f(\epsilon, \delta)$.

Suppose π is a probability function defined on the surface.

We define the sets B_1, B_2, B_3 to be inner trapezoids, each shares an edge with the triangle h and their upper bases form a smaller inner triangle s.t. $\pi(B_i) \ge \frac{\epsilon}{3}$.



If a dataset D visits each B_i, then we know that $err(h, C) = \pi(B_1 \cup B_2 \cup B_3) \le \epsilon$. We want to block the probability that D does not visit all B_is.
We know that: $\pi(D\ does\ not\ visit\ B_i) \le 1 - \frac{\epsilon}{3}$
Therefore: $\pi^m(D\ does\ not\ visit\ B_i) \le \left(1 - \frac{\epsilon}{3}\right)^m$.
Therefore:

$$\pi^m(D\ does\ not\ visit\ all\ B_i s) = \pi^m\left(\bigcup_{i=1}^{3} D\ does\ not\ visit\ B_i\right) \le \sum_{i=1}^{3}\left(1 - \frac{\epsilon}{3}\right)^m \le 3\exp\left(-\frac{m\epsilon}{3}\right) \le \delta$$

From that we get: $3\exp\left(-\frac{m\epsilon}{3}\right) \le \delta \Rightarrow ln(3\exp\left(-\frac{m\epsilon}{3}\right)) \le ln(\delta) \Rightarrow ln(3) + ln(\exp\left(-\frac{m\epsilon}{3}\right)) \le ln(\delta) \Rightarrow$ $ln(\exp\left(-\frac{m\epsilon}{3}\right))ln(\le ln(\frac{\delta}{3}) \Rightarrow -\frac{m\varepsilon}{3} \le ln(\frac{\delta}{3}) \Rightarrow m \ge \frac{3}{\varepsilon} ln(\frac{3}{\delta})$

Therefore, in order to know with probably δ that $err(h, C) \le \epsilon$, the training set size has to be at least $\frac{3}{\varepsilon} ln(\frac{3}{\delta})$.

4. The data we have is: $\alpha = 5,\ \hat{p} = 20, n = 1000$, hence:

   $p \in \left[\hat{p} \pm \Phi^{-1}\left(1-\frac{\alpha}{2}\right)\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}\right]$ with 95% confidence.

   $$p \in \left[0.2 \pm \Phi^{-1}(0.975)\sqrt{\frac{0.2 \cdot 0.8}{1000}}\right] \Rightarrow p \in [0.2 \pm 1.96 \cdot\ 0.01264911064] \Rightarrow p \in [0.2 \pm 0.02479]$$

   $$p\ \in [0.1752, 0.22479]$$

   The true error that can be expected is up to 22.479%

5.