Machine Learning from Data - Homework 5 (Theory) May-2022
IDs: [redacted]

1. Kernels and mapping functions (25 pts)

a. (20 pts) Let $K(x,y)=(x\cdot y+1)^3$ be a function over $\mathbb{R}^2\times\mathbb{R}^2$ (i.e., $x,y \in \mathbb{R}^2$).
Find ψ for which K is a kernel. (It may help to first expand the above term on the right-hand side).

$$K(x,y) = (x \cdot y + 1)^3 = (x^T y + 1)^3$$

$$x^T y = x_1 y_1 + x_2 y_2$$

$$K(x,y) = (x_1y_1 + x_2y_2 + 1)^3 = (x_1y_1 + x_2y_2 + 1)(x_1y_1 + x_2y_2 + 1)(x_1y_1 + x_2y_2 + 1)$$

Note: we will have 3 options choosing 3 options choosing 3 options, or $3^3 = 27$, and thus would expect the fully expanded form to have 27 total expressions (before combining like terms)

$$= x_1y_1^3 + x_1y_1^2 + x_2y_2 + x_1y_1^2 + x_1y_1^2x_2y_2 + x_1y_1x_2y_2^2 + x_1y_1x_2y_2 + x_1y_1^2 + x_1y_1x_2y_2 + x_1y_1$$

$$+x_1y_1x_2y_2^2 + x_2y_2^3 + x_2y_2^2 + x_1y_1^2x_2y_2 + x_1y_1x_2y_2^2 + x_1y_1x_2y_2 + x_1y_1x_2y_2 + x_2y_2^2 + x_2y_2$$

$$+x_1y_1^2 + x_1y_1x_2y_2 + x_1y_1 + x_1y_1x_2y_2 + x_2y_2^2 + x_2y_2 + x_1y_1 + x_2y_2 + 1$$

$$= x_1y_1^3 + x_2y_2^3 + 3x_1y_1^2 + 3x_2y_2^2 + 3x_1y_1 + 3x_2y_2 + 3x_1y_1^2x_2y_2 + 3x_1y_1x_2y_2^2 + 6x_1y_1x_2y_2 + 1$$

$$\psi(x_i) = \{x_1^3, \sqrt{3}x_1^2, \sqrt{3}x_1, \sqrt{3}x_2^2, \sqrt{3}x_2, \sqrt{3}x_1^2x_2, \sqrt{3}x_1x_2^2, \sqrt{6}x_1x_2, x_2^3, 1\}$$

$$K(x,y) = \psi(x_i)^T \cdot \psi(y_i)$$

$$= \{x_1^3, \sqrt{3}x_1^2, \sqrt{3}x_1, \sqrt{3}x_2^2, \sqrt{3}x_2, \sqrt{3}x_1^2x_2, \sqrt{3}x_1x_2^2, \sqrt{6}x_1x_2, x_2^3, 1\} \cdot$$
$$\{y_1^3, \sqrt{3}y_1^2, \sqrt{3}y_1, \sqrt{3}y_2^2, \sqrt{3}y_2, \sqrt{3}y_1^2y_2, \sqrt{3}y_1y_2^2, \sqrt{6}y_1y_2, y_2^3, 1\}$$

$$= (x^T y + 1)^3 = (x \cdot y + 1)^3 = K(x,y)$$

When the input space is $\mathbb{R}^2$, with 3rd-degree non-homogenous polynomial kernel, $d = 3, q = 2$ and the dimensionality of the feature space is given by:

$$m = \binom{d+q}{q} = \binom{3+2}{2} = 10$$

hence we have 10 terms in $\psi(x_i)$ and $\psi : \mathbb{R}^2 \to \mathbb{R}^{10}$

**reference for the above statement comes from: http://www.cs.rpi.edu/~stewart/lec23-post/kernels.pdf

b. (2 pts) What did we call the function ψ in class if we remove all coefficients?
(Full) rational varieties (of order 3)

c. (3 pts) How many multiplication operations do we save by using $K(x,y)$ versus $\psi(x)\cdot\psi(y)$?
Multiplication operations:
$\psi(x)\cdot\psi(y)$: 10 multiplications
$K(x,y)$: 2 multiplications + 2 multiplications = 4 multiplications
Saved: 10-4 = 6 multiplication operations saved

Just for fun - for ***all operations**** :
$\psi(x)\cdot\psi(y)$: 10 multiplications + 9 additions = 19 operations
$K(x,y)$: 2 multiplications + 1 additions + 2 multiplications = 5 operations
Saved: 19-5 = 14 total operations saved

2. Lagrange multipliers (25 pts)

Let $f(x,y)=2x-y$. Find the minimum and the maximum points for f under the constraint $g(x,y)=x^2/4+y^2=1$

$$f(x,y)=2x-y\ ;\ g(x,y):\frac{x^2}{4}+y^2=1$$

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

$$\frac{\partial}{\partial x}L(x,y)=2+\frac{2\lambda}{4}x=0\implies\frac{\lambda}{2}x=-2\implies x=\frac{-4}{\lambda}$$

$$\frac{\partial}{\partial y}L(x,y)=-1+2\lambda y=0\implies 2\lambda y=1\implies y=\frac{1}{2\lambda}$$

$$\frac{\partial}{\partial\lambda}L(x,y)=\frac{x^2}{4}+y^2-1=0\implies\frac{x^2}{4}+y^2=1\implies\frac{(\frac{-4}{\lambda})^2}{4}+(\frac{1}{2\lambda})^2=1\implies\frac{16}{4\lambda^2}+\frac{1}{4\lambda^2}=1$$

$$\lambda^2=\frac{17}{4}\implies\lambda=\sqrt{\frac{17}{4}}\sim 2.06155$$

$$L(x,y)_x=\frac{-4}{\lambda}=\frac{-4}{2.06155}\sim-1.940285\implies x=\pm1.94=\pm\frac{8}{\sqrt{17}}$$

$$L(x,y)_y=\frac{1}{2\lambda}=\frac{1}{4.1231}\sim 0.2425356\implies y=\pm0.24=\pm\frac{1}{\sqrt{17}}$$

which gives us the below extreme (x,y) coordinates to plug into $f(x,y)$:

$(1.94,-0.24)$, $(-1.94,0.24)$

$$f(1.94,-0.24)=2(1.94)-(-0.24)=4.12=\sqrt{17}\implies MAX$$

$$f(-1.94,0.24)=2(-1.94)-(0.24)=-4.12=-\sqrt{17}\implies MIN$$

3. PAC Learning (25 pts)

Let $X = \mathbb{R}^2$. Let vectors $u = \left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right), w = \left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right), v = (0, -1)$

****note:** correct $w = \left(\frac{-\sqrt{3}}{2}, \frac{1}{2}\right)$ ******

$$C = H = \left\{ h(r) = \left\{ (x_1, x_2) \middle| \begin{array}{l} (x_1, x_2) \cdot u \le r, \\ (x_1, x_2) \cdot v \le r, \\ (x_1, x_2) \cdot w \le r \end{array} \right\} \right\}, \text{ for } r > 0,$$

the set of all origin-centered upright equilateral triangles. Describe a polynomial sample complexity algorithm L that learns C using H. State the time complexity and the sample complexity of your suggested algorithm. Prove all your steps.

<u>Consistent Learner Algorithm:</u>

- Iterate over all instances {really only need to check all (+) instances}
- for each (+) instance, measure absolute value distance from origin and hold max_temp value
- max distance after iterating through all (+) instances - maximal distance to exactly touch furthest (+) instance(s) - will be the radii r, which when combined with the direction vectors *u, w, and v* and the constraints provided will form the lines and shape of the hypothesis origin-centered upright equilateral triangle, h ∈ H
- r* is the radius of the real concept target triangle where r <= r*
- Each sample drawn independently from an unknown distribution *D* will be assigned a (+) target value if it falls inside the triangle and (-) otherwise, and this true concept triangle with radius r* is what our learner is attempting to characterize.
- The drawing of the hypothesis triangle h ∈ H requires iterating through at most all *m* samples and is thus linear / polynomial algorithm, bounded by **O**(*m*) {or **O**(*m* * P(+)) = **O**(*m*)}
    - We will prove the sample complexity below, and show that *m* is polynomial in $\frac{1}{\epsilon}$ & $\frac{1}{\boldsymbol{\delta}}$
- Returns $h = L(D)$ s.t. $\forall x \in X$:
  $h(x) = 1 \Longrightarrow c(x) = 1$
- In our case, | H | = ∞ and isn't used explicitly for sample complexity bound on *m*, and instead *m* is bounded by $\frac{1}{\epsilon}$ & $\frac{1}{\boldsymbol{\delta}}$ as we show below
- We then have rϵ which is our "cutoff" radius - once r > rϵ, we're past the cutoff and probability < ϵ
- In the "good case", data from d ∈ D lands in Tr or our error region and r/h grows, and the perimeter (or difference between h & c) shrinks as our hypothesis converges towards c
- Probability of missing the T_r error region with probability of ϵ entirely is bounded by $\boldsymbol{\delta}$ - which we prove below along with *m* being polynomial in $\frac{1}{\epsilon}$ & $\frac{1}{\boldsymbol{\delta}}$ and thus that C is efficiently PAC-learnable
- We present a few crude illustrations of the efficient PAC learner below:

<u>Direct Calculation of Sample Complexity (plagiarized shamelessly from Recitation materials)</u>

We want Pr[$(x_1,x_2) \in T_r$] $\leq \epsilon$

$r^\epsilon = \text{arginf}_r$ Pr[(x1,x2) $\in$ Tr] $\leq \epsilon$ i.e. the largest perimeter triangle with probability at most ϵ

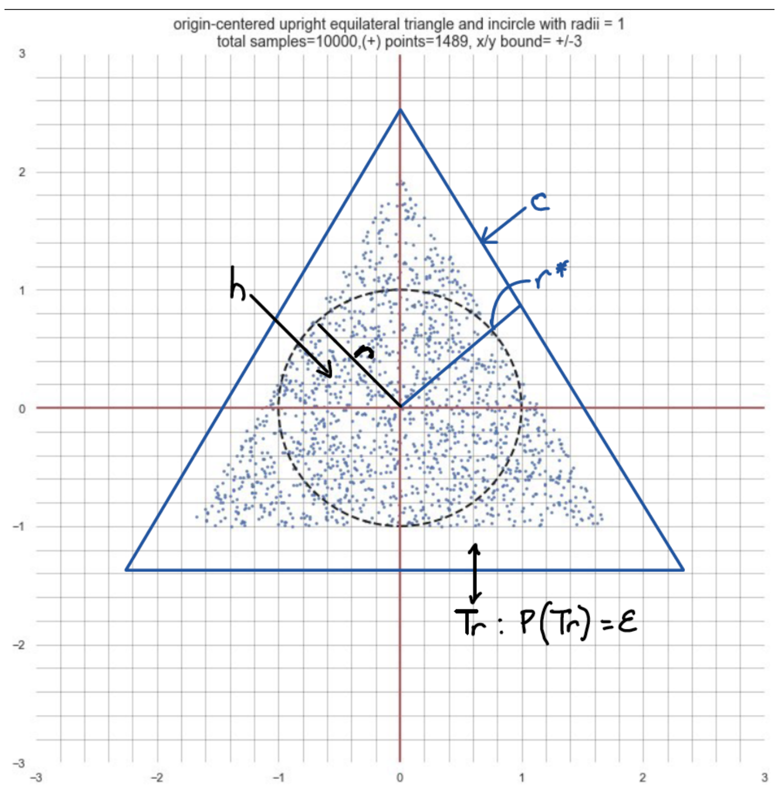
Case 1: if $r^\epsilon \leq r$ then probability of $T_r \leq \epsilon$

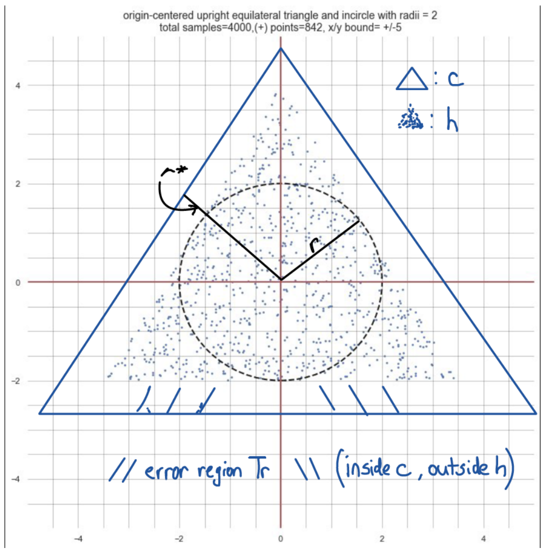
Case 2: probability of missing T_r with radii r^ϵ, r^* with *m* training samples?

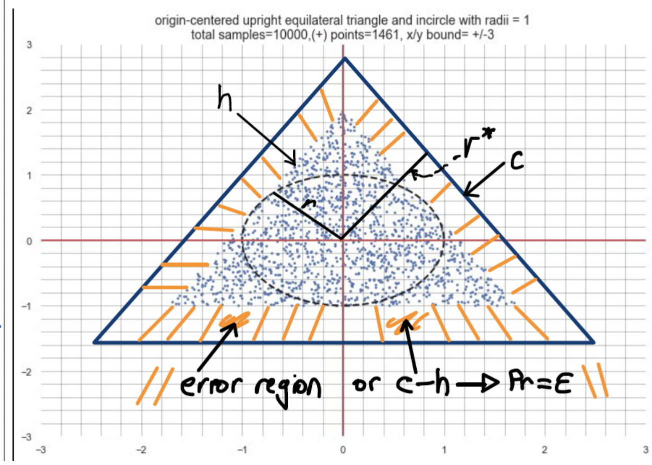
$(1-\epsilon)^m \leq \exp(-\epsilon m)$ and with sample size $m \geq \frac{ln(\frac{1}{\delta})}{\epsilon}$, we get:

$\exp(-\epsilon m) \leq \exp(\ -ln(\frac{1}{\delta})\ ) = \exp(\ln(\delta)) = \delta$

$$\text{for } \epsilon = 0.01,\ \delta = 0.05 \rightarrow m \geq 300 \text{ samples}$$
$$\text{for } \epsilon = 0.05,\ \delta = 0.05 \rightarrow m \geq 60 \text{ samples}$$
$$\text{for } \epsilon = 0.10,\ \delta = 0.01 \rightarrow m \geq 46 \text{ samples}$$
$$\text{for } \epsilon = 0.01,\ \delta = 0.001 \rightarrow m \geq 691 \text{ samples}$$







Discussion of perimeter T_r with $P(T_r)=\epsilon$ vs. 3 strips each with $P=\epsilon/3$:

- Notice we don't divide our error zone into chunks like with the rectangle where each strip accounted for P(strip) = $\epsilon/4$. For the origin-center equilateral triangle, it's like the annulus with the circle example in Rec10 - it's one single perimeter, denoted T_r , s.t. r <= r*, where r is the radius of our hypothesis h and r* is the radius of the real concept target triangle.
- T_r is the "error zone" or the difference / space between our hypothesis h generated by our learner *L(D)* and the true concept c.
  - Note this error region, the perimeter of symmetric difference between h & c, is limited to the perimeter triangle (T_r) that falls outside h but inside c. If we can guarantee that the probability under *D* of this perimeter is <= ϵ, then can have certainty of 1-$\boldsymbol{\delta}$ that the total error of h will be <= ϵ and that *L(D)* will yield an h that is ϵ-good
  - It's an important difference between the union-bound ϵ-subdivided model and the single "perimeter" or annulus = ϵ. This is because we only need one new (+) instance with maximal absolute value distance from the origin to shift or grow our entire h equilateral triangle outward towards our ϵ cutoff and in the direction of converging towards the true but unknown (to us) concept equilateral triangle.
  - The implication of using one perimeter vs. 3 or 4 slices is on the sample complexity bound. If we were to apply the formula using 3 equal-sized trapezoid-shaped chunks, we'd really just be doing more work than needed and would end up with a much looser and less precise sample complexity bound. For comparison, with ϵ & $\boldsymbol{\delta}$ = 0.05, the tighter bound with one single perimeter is m ≥ 60 samples, and using 3 chunks would yield a bound on m ≥ 246 samples. For ϵ & $\boldsymbol{\delta}$ = 0.01, it'd be 461 vs. 1712.

4. (15 pts) A business manager at your ecommerce company asked you to make a model to predict whether a user is going to proceed to checkout or abandon their cart. You created the model using, and reported 20% error on your test set of size 1000 samples. In the business manager's presentation to upper management, he presented your model and stated that the company can expect 20% error when deploying the model live on the website.
Luckily, you realize that this is a mistaken assumption, and you correct the statement to say that with 95% confidence, the true error they can expect is up to what percentage? (Just state the error percentage).

$95\%CI = \hat{p} \pm 2s.e.$ (or more correctly... $\hat{p} \pm 1.96s.e.$)

$\hat{p} = \frac{r}{n} = \frac{r-errors}{n-samples}$ ...$se = \sqrt{\frac{\hat{p}\cdot(1-\hat{p})}{n}}$

$\hat{p} = 20\% = 0.2$ $se = \sqrt{\frac{0.2\cdot(0.8)}{1000}} \sim 0.012649 \rightarrow 1.96se \sim 0.02479$

$95\%CI = 0.2 \pm 0.0248 = 0.1752 \longleftrightarrow 0.2248$

$\Longrightarrow$ thus upper bound of CI with 95% confidence $= 0.2248 = 22.48\%$

5. SVM (10 pts)

See the notebook in the homework files and follow the instructions there.
Take a **screenshot** of your resulting graph near the bottom of the notebook (titled "My Graph") and paste into your submission PDF along with your answers to the theoretical questions. Do **NOT** submit your code.