FINAL: 90 Minutes

Last Name: Solutions

First Name: _____

RIN: _____

Section: 4100 / 6100 (circle one)

Answer **ALL** questions.
NO COLLABORATION or electronic devices. Any violations result in an F.
NO questions allowed during the test. Interpret and do the best you can.
ALWAYS show your work and justify each answer.

GOOD LUCK!

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	TOTAL

- **You do not have time to waffle. So, don't waffle.**
- **Keep your answers precise and concise.**
- **Each question is worth 1 point.**

1. What is the two step approach to learning and why do we do it that way?

① $E_{in} \approx E_{out}$

② $E_{in} \approx 0$

Need to do this way because we don't know E_{out}

→ Can only get $E_{in} \approx 0$

→ Learning means $E_{out} \approx 0$

→ Need to link E_{in} to E_{out} in step 1.

2. You're a Facebook Troll. All your FB-friends have exciting lives based on their posts. In comparison, your life seems mundane and you get depressed. What learning from data trap could you be in?

Sampling bias or wrong bin

Bin 1 → Highlight Reel → estimate P[Exciting]

Bin 2 → Normal Reel → estimate P[Exciting]

Can't use P[Exciting] from bin 1 for bin 2.

3. Why is it a good idea to preprocess the data with input normalization?

To correct for arbitrary choices made during data collection such as units to measure (say) income.

4. Derive the growth function for the 1-dimensional positive intervals. Explain how to get the VC-dimension, and give the VC-dimension.

. . . . . n points

$\binom{N+1}{2}$ ways to place the two endpoints of the interval in two different spots.

Each gives a different dichotomy with at least 1 positive point.

Put both endpoints in same interval → all negative.

$$\therefore \boxed{m_{\mathcal{H}}(N) = 1 + \binom{N+1}{2}}$$

N	1	2	3
$m_{\mathcal{H}}(N)$	2	4	7

d_{vc} ↙ cheapest break point ↙

∴ VC-dimension is the max # points $\mathcal{H}$ can shatter

→ Cheapest break point − 1.

$$\boxed{d_{vc} = 2.}$$

5. Based on your VC-dimension for positive intervals, what is the error bar linking E_{in} to E_{out}?

$$E_{out} \leq E_{in} + O\left(\sqrt{\frac{d_{vc} \ln N}{N}}\right) \qquad d_{vc} = 2$$

error bar $O\left(\sqrt{\frac{2 \ln N}{N}}\right)$

More Exact.

$$E_{out} \leq E_{in} + \sqrt{\frac{8}{N} \ln 4\, m_{\mathcal{H}}(2N)}$$

$$\leq \sqrt{\frac{8}{N} d_{vc} \ln 2N + O(1)}$$

6. You decided to use 8th order polynomials to fit the data, to give your learning lots of flexibility. You suspect you might overfit the data. What does that mean?

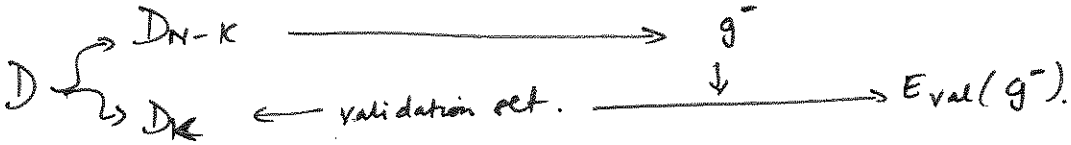
Model may fit noise in data (stochastic or deterministic) and thereby produce an inferior hypothesis with bad E_{out}.

7. In the previous problem, what could you do to help the learning in case of overfitting?

Regularize.

8. What is a validation set. Why do we use it?

Validation set gives peek at E_{out} with E_{val}.



Validation can be used to (for example) pick regularization parameter λ.

9. What are the tradeoffs in choosing the validation set? Why should it be small? Why large?

K large $\rightarrow$ E_{val} is more accurate

K small $\rightarrow$ $g^- \approx g$ $\therefore$ $E_{val}(g)$ is a better estimate of $E_{out}(g)$

10. Define the leave one out cross-validation error, E_{CV}?

$$E_{CV} = \frac{1}{N} \sum_{i=1}^{N} e_i(g_i^-(x_i), y_i).$$

where g_i^- is the deficient hypothesis obtained from data minus (x_i, y_i).

e_i is your error function

$e_i(g_i^-(x_i), y_i)$ is the error between your deficient predictor on the left out point.

11. Prove that E_{CV} based on N data points is an unbiased estimate of your expected out-of-sample error when you learn on $N-1$ data points.?

$$E_D[E_{CV}] = E_D\left[\frac{1}{N}\sum_{i=1}^{N} e_i(g_i^-(x_i), y_i).\right]$$

$$= \frac{1}{N}\sum_{i=1}^{N} E_D\, e_i(g_i^-(x_i), y_i)$$

$$= \frac{1}{N}\sum_{i=1}^{N} E_{D_i^-}\left[E_{x_i,y_i}[e_i]\right]$$

$E_{out}(g_i^-)$.

$E(N-1)$.

$$= \frac{1}{N}\sum_{i=1}^{N} E(N-1) = E(N-1).$$

12. The nearest neighbor algorithm was at most a factor of 2 away from optimal. Prove that the 3-nearest neighbor algorithm is near-optimal. That is, for large enough N, the out of sample error at a test point x ia bounded by

$$E_{\text{out}}(\text{3-NN}) \leq E^*_{\text{out}}(1 + 3E^*_{\text{out}}).$$

If optimal performance is 1% error, how bad can 3-NN be? (In your proof, state any assumptions.)

Assume. $\pi(x) = P[y=1 \mid x]$ is smooth

$x_{[1]}\ x_{[2]}\ x_{[3]} \longrightarrow x$ as $N \longrightarrow \infty$.

$\pi(x_{[i]}) \longrightarrow \pi(x)$

$P[\text{error}] = P[\text{error} \mid y=1]\,\pi(x) + P[\text{error} \mid y=-1](1-\pi(x))$.

$3(1-\pi)^2\pi + (1-\pi)^3 \qquad 3\pi^2(1-\pi) + \pi^3$.

↖ symmetric in $\pi, 1-\pi$

$E^*_{\text{out}} = \min(\pi, 1-\pi)$ ∴ we can replace π by E^*_{out}.

$P[\text{error}] = \pi(1-\pi)\left[3(1-\pi)\pi + (1-\pi)^3\right] + \pi(1-\pi)\left[3\pi(1-\pi) + \pi^2\right]$

$= \pi(1-\pi)\left[6\pi(1-\pi) + (1-\pi)^2 + \pi^2\right]$

$= \pi(1-\pi)\left[4\pi(1-\pi) + 1\right]$

$= E^*_{\text{out}}(1-E^*_{\text{out}})\left[4E^*_{\text{out}}(1-E^*_{\text{out}}) + 1\right]$

$= E^*_{\text{out}}\left(1 + 3E^*_{\text{out}} - 4(E^*_{\text{out}})^2 + 4E^{*3}_{\text{out}}\right)$ $\qquad E^*_{\text{out}} \leq \frac{1}{2}$

$\leq E^*_{\text{out}}(1 + 3E^*_{\text{out}})$.

$E^*_{\text{out}} = 0.01 \rightarrow P[\text{error}] \leq 0.01 \cdot (1.03)$

$= 0.0103 = 1.03\%$.

3-NN asymptotically has $E_{\text{out}} \leq 1.03\%$, pretty close to optimal.

3-NN is enough!

13. Explain why the 1-hidden layer neural network is more powerful than the K-RBF network. What are the pros and cons of this power?

K-RBF Network: $g(x) = \text{sign}\left(w_0 + \sum_{i=1}^{K} w_i \phi\left(\frac{x - \mu_i}{r}\right)\right)$ ← similarity features fixed via unsupervised learning

NN: $g(x) = \text{sign}\left(w_0 + \sum_{i=1}^{m} w_i \theta(v_i^T x)\right)$

$\theta(v_i^T x)$: fully tunable features, v_i and w_i are jointly learned.

KRBF network: μ_j → unsupervised → linear model given μ_j

NN: fully tuned features → more powerful.

14. Explain why the optimal hyperplane with maximum margin performs well even in very high dimensional spaces where a random perceptron won't.

$E_{CV} \approx E_{out}(N-1)$ is controlled by # support vectors which can be small even in high dimensions.

15. Why is it computationally tractable to run the optimal hyperplane using a feature transform into very high, even infinite, dimensions.

In dual space it is an Inner Product Algorithm

→ can work in original X-space if given the kernel that computes the inner product.

↑

Never have to go to the infinite feature space.