FINAL: 90 Minutes

Last Name: Solutions

First Name: _____

RIN: _____

Section: 4100 / 6100

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.

GOOD LUCK!

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

- **You do not have time to waffle.**
- **Keep your answers precise and concise.**

1. "Machine learning is like walking forward while looking backward." Explain.

Looking backward is the data (the past)
Walking forward is where you need to predict (the future; the test point)
Using the data you have to predict the test point to succeed.

2. What is *learning from data*?

Given data, output a predictor g (final hypothesis) which performs better than random, preferably obtaining near optimal E_{out}

ALL WE CARE ABOUT IS E_{OUT}

3. What is the 2-step approach to learning and why do we do it that way?

① Ensure $E_{out} \approx E_{in}$

② Get E_{in} as small as possible

We care about E_{out}, but we cant measure E_{out}
We can measure E_{in} so we optimize what we can measure
While ensuring theoretically step ①.

4. Explain why no hypothesis set can have growth function $m_{\mathcal{H}}(N) = 1 + N + \frac{1}{6}N(N-1)(N-2)$.

$m(2) = 3 < 2^2 \therefore 2$ is a break point

<u>Theorem</u> if 2 is a break point then $m(N) \leq \binom{N}{0} + \binom{N}{1} + \binom{N}{2}$

Our $m(N) = \binom{N}{0} + \binom{N}{1} + \binom{N}{3}$ our $m(N)$ does not satisfy the bound in the theorem - For example try $N = 10$ $\binom{10}{3} > \binom{10}{2}$.

5. Why is it important to learn using a hypothesis set with *finite* VC-dimension?

finite d_{vc} means $E_{out} \leq E_{in} + \underbrace{\text{ERROR BAR}}_{\to 0 \text{ when } N \to \infty}$

$\therefore$ with large enough N we ensure step 1 of learning and we can go ahead and safely minimize E_{in} within $\mathcal{H}$.

6. "We are regularized by our parents." Explain.

we are learning machines from birth, learning to act optimally within our <u>noisy</u> environment.

Regularization is necessary.

The parents constrain the learning in the "right direction" to help us not to overfit.

<u>Otherwise</u> (for example) children would just eat candy [feels good $\Leftrightarrow$ low E_{in}].

7. What is overfitting?

Fitting the data more than is warranted, where decreasing E_{in} leads to increase in E_{out}.

8. What are the causes of overfitting?

<u>Noise</u>:

$\underbrace{\text{Measurement error}}_{\text{stochastic noise}}$

$\underbrace{\text{Too complex } f \text{ for } \mathcal{H}}_{\text{deterministic noise}}$

9. What are the tools for curing overfitting?

Regularization
↓
Constrain the learning

Validation
↓
Estimate E_{out} to pick models with best E_{out}

10. A manager asked the ML-expert to solve an ML-problem but witheld 500 data points. Discuss the pros and cons of the manager's action.

Pros: Manager does not need to "trust" the ML-expert. Manager will be able to estimate E_{out} (unbiased)

Cons: ML-expert has fewer data to learn on and so will produce inferior hypotheses.

11. How does the approximation-generalization tradeoff impact your choice of $\mathcal{H}$?

To get good generalization: $E_{in} \approx E_{out}$ you choose small $\mathcal{H}$.

To get good approximation: Choose big $\mathcal{H}$.

N is the primary factor which determines the "size" of $\mathcal{H}$.

12. For the Nearest Neighbor classifier, what is E_{in}? Is $E_{\text{in}} \approx E_{\text{out}}$?

$E_{in} = 0$

E_{out} can be anything so $E_{in} \not\approx E_{out}$.

13. How do you reconcile the Nearest Neighbor's E_{in} with the two-step approach to learning?

NN does not fit into the 2-step approach.
All we care about is E_{out} so we need to give some other guarantee on E_{out}, other than $E_{in} \approx E_{out}$.
In fact we proved for $N \to \infty$ $E_{out} \leq 2 E_{out}^*$

14. What is the difference between the perceptron and the k-RBF network? Give the formulas for the final hypothesis (classification) and refer to them in explaining the difference.

perceptron: $h(x) = \text{sign}(w^T x)$
$= \text{sign}\left(\underbrace{\sum_{i=1}^{d} w_i x_i + w_0}_{\text{score based.}}\right)$

k-RBF: $h(x) = \text{sign}\left(\sum_{j=1}^{k} w_j \phi\left(\frac{x-\mu_j}{r}\right) + w_0\right)$

perceptron with similarity features.
technically non-linear in μ_j
μ_j fixed by Lloyds algorithm.

15. What is the difference between the perceptron+PLA and the support vector machine? What optimization problem (separable case) does the SVM solve and how do you solve it?

perceptron+PLA $\longrightarrow$ any separating hyperplane

SVM $\longrightarrow$ optimal, most robust to noise, hyperplane.

SVM $\min_{w,b} \frac{1}{2} w^T w$
s.t. $y_n(w^T x_n + b) \geq 1 \quad \forall n = 1, 2, \ldots N.$

16. Explain how/why we are able to use the support vector machine *efficiently* with an infinite dimensional feature and also have control over E_{out}?

Infinite dimensional feature transform implemented using the kernel, so its efficient.

$E_{out} \leq E_{cv} \leq \frac{\# SV}{N}$ so E_{out} is controlled by a parameter not explicitly related to dimension

in infinite dim, it is possible to have a small # of SV's.