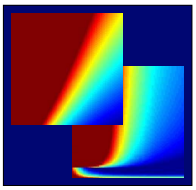
Learning From Data
Yaser Abu-Mostafa, *Caltech*
http://work.caltech.edu/telecourse
Self-paced version



Homework # 1

All questions have multiple-choice answers ([a], [b], [c], ...*). You can collaborate with others, but do not discuss the selected or excluded choices in the answers. You can consult books and notes, but not other people's solutions. Your solutions should be based on your own work. Definitions and notation follow the lectures.*

Note about the homework

- The goal of the homework is to facilitate a deeper understanding of the course material. The questions are not designed to be puzzles with catchy answers. They are meant to make you roll up your sleeves, face uncertainties, and approach the problem from different angles.
- The problems range from easy to difficult, and from practical to theoretical. Some problems require running a full experiment to arrive at the answer.
- The answer may not be obvious or numerically close to one of the choices, but one (and only one) choice will be correct if you follow the instructions precisely in each problem. You are encouraged to explore the problem further by experimenting with variations on these instructions, for the learning benefit.
- You are also encouraged to take part in the forum

  http://book.caltech.edu/bookforum

  where there are many threads about each homework set. We hope that you will contribute to the discussion as well. Please follow the forum guidelines for posting answers (see the "BEFORE posting answers" announcement at the top there).

- **The Learning Problem**

1. What types of Machine Learning, if any, best describe the following three scenarios:

(i) A coin classification system is created for a vending machine. The developers obtain exact coin specifications from the U.S. Mint and derive a statistical model of the size, weight, and denomination, which the vending machine then uses to classify coins.

(ii) Instead of calling the U.S. Mint to obtain coin information, an algorithm is presented with a large set of labeled coins. The algorithm uses this data to infer decision boundaries which the vending machine then uses to classify its coins.

(iii) A computer develops a strategy for playing Tic-Tac-Toe by playing repeatedly and adjusting its strategy by penalizing moves that eventually lead to losing.

[a] (i) Supervised Learning, (ii) Unsupervised Learning, (iii) Reinforcement Learning

[b] (i) Supervised Learning, (ii) Not learning, (iii) Unsupervised Learning

[c] (i) Not learning, (ii) Reinforcement Learning, (iii) Supervised Learning

[d] (i) Not learning, (ii) Supervised Learning, (iii) Reinforcement Learning

[e] (i) Supervised Learning, (ii) Reinforcement Learning, (iii) Unsupervised Learning

2. Which of the following problems are best suited for Machine Learning?

(i) Classifying numbers into primes and non-primes.

(ii) Detecting potential fraud in credit card charges.

(iii) Determining the time it would take a falling object to hit the ground.

(iv) Determining the optimal cycle for traffic lights in a busy intersection.

[a] (ii) and (iv)

[b] (i) and (ii)

[c] (i), (ii), and (iii)

[d] (iii)

[e] (i) and (iii)

- **Bins and Marbles**

3. We have 2 opaque bags, each containing 2 balls. One bag has 2 black balls and the other has a black ball and a white ball. You pick a bag at random and then pick one of the balls in that bag at random. When you look at the ball, it is black. You now pick the second ball from that same bag. What is the probability that this ball is also black?

[a] 1/4

[b] 1/3

[c] 1/2

[d] 2/3

[e] 3/4

Consider a sample of 10 marbles drawn from a bin containing red and green marbles. The probability that any marble we draw is red is $\mu = 0.55$ (independently, with replacement). We address the probability of getting no red marbles ($\nu = 0$) in the following cases:

4. We draw only one such sample. Compute the probability that $\nu = 0$. The closest answer is ('closest answer' means: $|\text{your answer} - \text{given option}|$ is closest to 0):

[a] 7.331×10^{-6}

[b] 3.405×10^{-4}

[c] 0.289

[d] 0.450

[e] 0.550

5. We draw 1,000 independent samples. Compute the probability that (at least) one of the samples has $\nu = 0$. The closest answer is:

[a] 7.331×10^{-6}

[b] 3.405×10^{-4}

[c] 0.289

[d] 0.450

[e] 0.550

• **Feasibility of Learning**

Consider a Boolean target function over a 3-dimensional input space $\mathcal{X} = \{0, 1\}^3$ (instead of our ± 1 binary convention, we use 0,1 here since it is standard for Boolean functions). We are given a data set $\mathcal{D}$ of five examples represented in the table below, where $y_n = f(\mathbf{x}_n)$ for $n = 1, 2, 3, 4, 5$.

$\mathbf{x}_n$			y_n
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1

Note that in this simple Boolean case, we can enumerate the entire input space (since there are only $2^3 = 8$ distinct input vectors), and we can enumerate the set of all possible target functions (there are only $2^{2^3} = 256$ distinct Boolean function on 3 Boolean inputs).
Let us look at the problem of learning f. Since f is unknown except inside $\mathcal{D}$, any function that agrees with $\mathcal{D}$ could conceivably be f. Since there are only 3 points in $\mathcal{X}$ outside $\mathcal{D}$, there are only $2^3 = 8$ such functions.
The remaining points in $\mathcal{X}$ which are not in $\mathcal{D}$ are: 101, 110, and 111. We want to determine the hypothesis that agrees the most with the possible target functions. In order to quantify this, count how many of the 8 possible target functions agree with each hypothesis on all 3 points, how many agree on just 2 of the points, on just 1 point, and how many do not agree on any points. The final score for each hypothesis is computed as follows:

Score = *(# of target functions agreeing with hypothesis on all 3 points)*×3 + *(# of target functions agreeing with hypothesis on exactly 2 points)*×2 + *(# of target functions agreeing with hypothesis on exactly 1 point)*×1 + *(# of target functions agreeing with hypothesis on 0 points)*×0.

6. Which hypothesis g agrees the most with the possible target functions in terms of the above score?

[a] g returns 1 for all three points.

[b] g returns 0 for all three points.

[c] g is the XOR function applied to $\mathbf{x}$, i.e., if the number of 1s in $\mathbf{x}$ is odd, g returns 1; if it is even, g returns 0.

[d] g returns the opposite of the XOR function: if the number of 1s is odd, it returns 0, otherwise returns 1.

[e] They are all equivalent (equal scores for g in **[a]** through **[d]**).

• **The Perceptron Learning Algorithm**

In this problem, you will create your own target function f and data set $\mathcal{D}$ to see how the Perceptron Learning Algorithm works. Take $d = 2$ so you can visualize the problem, and assume $\mathcal{X} = [-1, 1] \times [-1, 1]$ with uniform probability of picking each $\mathbf{x} \in \mathcal{X}$.

In each run, choose a random line in the plane as your target function f (do this by taking two random, uniformly distributed points in $[-1, 1] \times [-1, 1]$ and taking the line passing through them), where one side of the line maps to $+1$ and the other maps to -1. Choose the inputs $\mathbf{x}_n$ of the data set as random points (uniformly in $\mathcal{X}$), and evaluate the target function on each $\mathbf{x}_n$ to get the corresponding output y_n.

Now, in each run, use the Perceptron Learning Algorithm to find g. Start the PLA with the weight vector $\mathbf{w}$ being all zeros (consider $\text{sign}(0) = 0$, so all points are initially misclassified), and at each iteration have the algorithm choose a point randomly from the set of misclassified points. We are interested in two quantities: the number of iterations that PLA takes to converge to g, and the disagreement between f and g which is $\mathbb{P}[f(\mathbf{x}) \neq g(\mathbf{x})]$ (the probability that f and g will disagree on their classification of a random point). You can either calculate this probability exactly, or approximate it by generating a sufficiently large, separate set of points to estimate it.

In order to get a reliable estimate for these two quantities, you should repeat the experiment for 1000 runs (each run as specified above) and take the average over these runs.

7. Take $N = 10$. How many iterations does it take on average for the PLA to converge for $N = 10$ training points? Pick the value closest to your results (again, 'closest' means: |your answer − given option| is closest to 0).

[a] 1

[b] 15

[c] 300

[d] 5000

[e] 10000

8. Which of the following is closest to $\mathbb{P}[f(\mathbf{x}) \neq g(\mathbf{x})]$ for $N = 10$?

[a] 0.001

[b] 0.01

[c] 0.1

[d] 0.5

[e] 0.8

9. Now, try $N = 100$. How many iterations does it take on average for the PLA to converge for $N = 100$ training points? Pick the value closest to your results.

[a] 50

[b] 100

[c] 500

[d] 1000

[e] 5000

10. Which of the following is closest to $\mathbb{P}[f(\mathbf{x}) \neq g(\mathbf{x})]$ for $N = 100$?

[a] 0.001

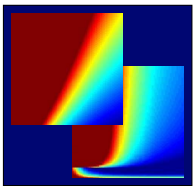
[b] 0.01

[c] 0.1

[d] 0.5

[e] 0.8

Learning From Data
Yaser Abu-Mostafa, *Caltech*
http://work.caltech.edu/telecourse
Self-paced version



Homework # 2

*All questions have multiple-choice answers (***[a], [b], [c], ...***). You can collaborate with others, but do not discuss the selected or excluded choices in the answers. You can consult books and notes, but not other people's solutions. Your solutions should be based on your own work. Definitions and notation follow the lectures.*

Note about the homework

- The goal of the homework is to facilitate a deeper understanding of the course material. The questions are not designed to be puzzles with catchy answers. They are meant to make you roll up your sleeves, face uncertainties, and approach the problem from different angles.
- The problems range from easy to difficult, and from practical to theoretical. Some problems require running a full experiment to arrive at the answer.
- The answer may not be obvious or numerically close to one of the choices, but one (and only one) choice will be correct if you follow the instructions precisely in each problem. You are encouraged to explore the problem further by experimenting with variations on these instructions, for the learning benefit.
- You are also encouraged to take part in the forum

  http://book.caltech.edu/bookforum

  where there are many threads about each homework set. We hope that you will contribute to the discussion as well. Please follow the forum guidelines for posting answers (see the "BEFORE posting answers" announcement at the top there).

- **Hoeffding Inequality**

Run a computer simulation for flipping 1,000 virtual fair coins. Flip each coin independently 10 times. Focus on 3 coins as follows: c_1 is the first coin flipped, c_{rand} is a coin chosen randomly from the 1,000, and c_{min} is the coin which had the minimum frequency of heads (pick the earlier one in case of a tie). Let ν_1, ν_{rand}, and ν_{min} be the *fraction* of heads obtained for the 3 respective coins out of the 10 tosses.

Run the experiment 100,000 times in order to get a full distribution of ν_1, ν_{rand}, and ν_{min} (note that c_{rand} and c_{min} will change from run to run).

1. The average value of ν_{min} is closest to:

[a] 0

[b] 0.01

[c] 0.1

[d] 0.5

[e] 0.67

2. Which coin(s) has a distribution of ν that satisfies the (single-bin) Hoeffding Inequality?

[a] c_1 only

[b] c_{rand} only

[c] c_{min} only

[d] c_1 and c_{rand}

[e] c_{min} and c_{rand}

- **Error and Noise**

Consider the bin model for a hypothesis h that makes an error with probability μ in approximating a deterministic target function f (both h and f are binary functions). If we use the same h to approximate a noisy version of f given by:

$$P(y \mid \mathbf{x}) = \begin{cases} \lambda & y = f(x) \\ 1 - \lambda & y \neq f(x) \end{cases}$$

3. What is the probability of error that h makes in approximating y? *Hint: Two wrongs can make a right!*

[a] μ

[b] λ

[c] 1-μ

[d] $(1-\lambda)*\mu+\lambda*(1-\mu)$

[e] $(1-\lambda)*(1-\mu)+\lambda*\mu$

4. At what value of λ will the performance of h be independent of μ?

[a] 0

[b] 0.5

[c] $1/\sqrt{2}$

[d] 1

[e] No values of λ

• Linear Regression

In these problems, we will explore how Linear Regression for classification works. As with the Perceptron Learning Algorithm in Homework # 1, you will create your own target function f and data set $\mathcal{D}$. Take $d = 2$ so you can visualize the problem, and assume $\mathcal{X} = [-1, 1] \times [-1, 1]$ with uniform probability of picking each $\mathbf{x} \in \mathcal{X}$. In each run, choose a random line in the plane as your target function f (do this by taking two random, uniformly distributed points in $[-1, 1] \times [-1, 1]$ and taking the line passing through them), where one side of the line maps to $+1$ and the other maps to -1. Choose the inputs $\mathbf{x}_n$ of the data set as random points (uniformly in $\mathcal{X}$), and evaluate the target function on each $\mathbf{x}_n$ to get the corresponding output y_n.

5. Take $N = 100$. Use Linear Regression to find g and evaluate E_{in}, the fraction of in-sample points which got classified incorrectly. Repeat the experiment 1000 times and take the average (keep the g's as they will be used again in Problem 6). Which of the following values is closest to the average E_{in}? (*Closest* is the option that makes the expression |your answer − given option| closest to 0. Use this definition of *closest* here and throughout.)

[a] 0

[b] 0.001

[c] 0.01

[d] 0.1

[e] 0.5

6. Now, generate 1000 fresh points and use them to estimate the out-of-sample error E_{out} of g that you got in Problem 5 (number of misclassified out-of-sample points / total number of out-of-sample points). Again, run the experiment 1000 times and take the average. Which value is closest to the average E_{out}?

[a] 0

[b] 0.001

[c] 0.01

[d] 0.1

[e] 0.5

7. Now, take $N = 10$. After finding the weights using Linear Regression, use them as a vector of initial weights for the Perceptron Learning Algorithm. Run PLA until it converges to a final vector of weights that completely separates all the in-sample points. Among the choices below, what is the closest value to the average number of iterations (over 1000 runs) that PLA takes to converge? (When implementing PLA, have the algorithm choose a point randomly from the set of misclassified points at each iteration)

[a] 1

[b] 15

[c] 300

[d] 5000

[e] 10000

• Nonlinear Transformation

In these problems, we again apply Linear Regression for classification. Consider the target function:

$$f(x_1, x_2) = \text{sign}(x_1^2 + x_2^2 - 0.6)$$

Generate a training set of $N = 1000$ points on $\mathcal{X} = [-1, 1] \times [-1, 1]$ with a uniform probability of picking each $\mathbf{x} \in \mathcal{X}$. Generate simulated noise by flipping the sign of the output in a randomly selected 10% subset of the generated training set.

8. Carry out Linear Regression without transformation, i.e., with feature vector:

$$(1, x_1, x_2),$$

to find the weight $\mathbf{w}$. What is the closest value to the classification in-sample error E_{in}? (Run the experiment 1000 times and take the average E_{in} to reduce variation in your results.)

[a] 0

[b] 0.1

[c] 0.3

[d] 0.5

[e] 0.8

9. Now, transform the $N = 1000$ training data into the following nonlinear feature vector:

$$(1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$$

Find the vector $\tilde{\mathbf{w}}$ that corresponds to the solution of Linear Regression. Which of the following hypotheses is closest to the one you find? Closest here means agrees the most with your hypothesis (has the highest probability of agreeing on a randomly selected point). Average over a few runs to make sure your answer is stable.

[a] $g(x_1, x_2) = \text{sign}(-1 - 0.05x_1 + 0.08x_2 + 0.13x_1x_2 + 1.5x_1^2 + 1.5x_2^2)$

[b] $g(x_1, x_2) = \text{sign}(-1 - 0.05x_1 + 0.08x_2 + 0.13x_1x_2 + 1.5x_1^2 + 15x_2^2)$

[c] $g(x_1, x_2) = \text{sign}(-1 - 0.05x_1 + 0.08x_2 + 0.13x_1x_2 + 15x_1^2 + 1.5x_2^2)$

[d] $g(x_1, x_2) = \text{sign}(-1 - 1.5x_1 + 0.08x_2 + 0.13x_1x_2 + 0.05x_1^2 + 0.05x_2^2)$

[e] $g(x_1, x_2) = \text{sign}(-1 - 0.05x_1 + 0.08x_2 + 1.5x_1x_2 + 0.15x_1^2 + 0.15x_2^2)$

10. What is the closest value to the classification out-of-sample error E_{out} of your hypothesis from Problem 9? (Estimate it by generating a new set of 1000 points and adding noise, as before. Average over 1000 runs to reduce the variation in your results.)

[a] 0

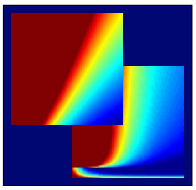
[b] 0.1

[c] 0.3

[d] 0.5

[e] 0.8

Learning From Data
Yaser Abu-Mostafa, *Caltech*
http://work.caltech.edu/telecourse
Self-paced version



Homework # 3

*All questions have multiple-choice answers (***[a], [b], [c], ...***). You can collaborate with others, but do not discuss the selected or excluded choices in the answers. You can consult books and notes, but not other people's solutions. Your solutions should be based on your own work. Definitions and notation follow the lectures.*

Note about the homework

- The goal of the homework is to facilitate a deeper understanding of the course material. The questions are not designed to be puzzles with catchy answers. They are meant to make you roll up your sleeves, face uncertainties, and approach the problem from different angles.
- The problems range from easy to difficult, and from practical to theoretical. Some problems require running a full experiment to arrive at the answer.
- The answer may not be obvious or numerically close to one of the choices, but one (and only one) choice will be correct if you follow the instructions precisely in each problem. You are encouraged to explore the problem further by experimenting with variations on these instructions, for the learning benefit.
- You are also encouraged to take part in the forum

  http://book.caltech.edu/bookforum

  where there are many threads about each homework set. We hope that you will contribute to the discussion as well. Please follow the forum guidelines for posting answers (see the "BEFORE posting answers" announcement at the top there).

- **Generalization Error**

1. The modified Hoeffding Inequality provides a way to characterize the generalization error with a probabilistic bound

$$\mathbb{P}\left[|E_{in}(g) - E_{out}(g)| > \epsilon\right] \leq 2Me^{-2\epsilon^2 N}$$

for any $\epsilon > 0$. If we set $\epsilon = 0.05$ and want the probability bound $2Me^{-2\epsilon^2 N}$ to be at most 0.03, what is the least number of examples N (among the given choices) needed for the case $M = 1$?

[a] 500

[b] 1000

[c] 1500

[d] 2000

[e] More examples are needed.

2. Repeat for the case $M = 10$.

[a] 500

[b] 1000

[c] 1500

[d] 2000

[e] More examples are needed.

3. Repeat for the case $M = 100$.

[a] 500

[b] 1000

[c] 1500

[d] 2000

[e] More examples are needed.

- **Break Point**

4. As shown in class, the (smallest) break point for the Perceptron Model in the two-dimensional case ($\mathbb{R}^2$) is 4 points. What is the smallest break point for the Perceptron Model in $\mathbb{R}^3$? (i.e., instead of the hypothesis set consisting of separating lines, it consists of separating planes.)

[a] 4

[b] 5

[c] 6

[d] 7

[e] 8

- **Growth Function**

5. Which of the following are possible formulas for a growth function $m_{\mathcal{H}}(N)$:

$$\begin{array}{ll} \text{i) } 1+N & \text{iv) } 2^{\lfloor N/2 \rfloor} \\ \text{ii) } 1+N+\binom{N}{2} & \text{v) } 2^N \\ \text{iii) } \sum_{i=1}^{\lfloor \sqrt{N} \rfloor} \binom{N}{i} & \end{array}$$

where $\lfloor u \rfloor$ is the biggest integer $\leq u$, and $\binom{M}{m} = 0$ when $m > M$.

[a] i, v

[b] i, ii, v

[c] i, iv, v

[d] i, ii, iii, v

[e] i, ii, iii, iv, v

- **Fun with Intervals**

6. Consider the "2-intervals" learning model, where h: $\mathbb{R} \to \{-1, +1\}$ and $h(x) = +1$ if the point is within either of two arbitrarily chosen intervals and -1 otherwise. What is the (smallest) break point for this hypothesis set?

[a] 3

[b] 4

[c] 5

[d] 6

[e] 7

7. Which of the following is the growth function $m_H(N)$ for the "2-intervals" hypothesis set?

[a] $\binom{N+1}{4}$

[b] $\binom{N+1}{2} + 1$

[c] $\binom{N+1}{4} + \binom{N+1}{2} + 1$

[d] $\binom{N+1}{4} + \binom{N+1}{3} + \binom{N+1}{2} + \binom{N+1}{1} + 1$

[e] None of the above

8. Now, consider the general case: the "M-intervals" learning model. Again $h : \mathbb{R} \to \{-1, +1\}$, where $h(x) = +1$ if the point falls inside any of M arbitrarily chosen intervals, otherwise $h(x) = -1$. What is the (smallest) break point of this hypothesis set?

[a] M

[b] $M + 1$

[c] M^2

[d] $2M + 1$

[e] $2M - 1$

- **Convex Sets: The Triangle**

9. Consider the "triangle" learning model, where $h : \mathbb{R}^2 \to \{-1, +1\}$ and $h(\mathbf{x}) = +1$ if $\mathbf{x}$ lies within an arbitrarily chosen triangle in the plane and -1 otherwise. Which is the largest number of points in $\mathbb{R}^2$ (among the given choices) that can be shattered by this hypothesis set?

[a] 1

[b] 3

[c] 5

[d] 7

[e] 9

- **Non-Convex Sets: Concentric Circles**

10. Compute the growth function $m_{\mathcal{H}}(N)$ for the learning model made up of two concentric circles in $\mathbb{R}^2$. Specifically, $\mathcal{H}$ contains the functions which are $+1$ for

$$a^2 \leq x_1^2 + x_2^2 \leq b^2$$

and -1 otherwise. The growth function is

[a] $N + 1$

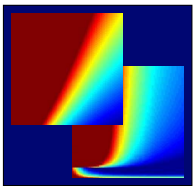
[b] $\binom{N+1}{2} + 1$

[c] $\binom{N+1}{3} + 1$

[d] $2N^2 + 1$

[e] None of the above

Learning From Data
Yaser Abu-Mostafa, *Caltech*
http://work.caltech.edu/telecourse
Self-paced version



Homework # 4

*All questions have multiple-choice answers (***[a], [b], [c], ...***). You can collaborate with others, but do not discuss the selected or excluded choices in the answers. You can consult books and notes, but not other people's solutions. Your solutions should be based on your own work. Definitions and notation follow the lectures.*

Note about the homework

- The goal of the homework is to facilitate a deeper understanding of the course material. The questions are not designed to be puzzles with catchy answers. They are meant to make you roll up your sleeves, face uncertainties, and approach the problem from different angles.
- The problems range from easy to difficult, and from practical to theoretical. Some problems require running a full experiment to arrive at the answer.
- The answer may not be obvious or numerically close to one of the choices, but one (and only one) choice will be correct if you follow the instructions precisely in each problem. You are encouraged to explore the problem further by experimenting with variations on these instructions, for the learning benefit.
- You are also encouraged to take part in the forum

  http://book.caltech.edu/bookforum

  where there are many threads about each homework set. We hope that you will contribute to the discussion as well. Please follow the forum guidelines for posting answers (see the "BEFORE posting answers" announcement at the top there).

- **Generalization Error**

In Problems 1-3, we look at generalization bounds numerically. For $N > d_{\text{VC}}$, use the simple approximate bound $N^{d_{\text{VC}}}$ for the growth function $m_{\mathcal{H}}(N)$.

1. For an $\mathcal{H}$ with $d_{\text{VC}} = 10$, if you want 95% confidence that your generalization error is at most 0.05, what is the closest numerical approximation of the sample size that the VC generalization bound predicts?

[a] 400,000

[b] 420,000

[c] 440,000

[d] 460,000

[e] 480,000

2. There are a number of bounds on the generalization error ϵ, all holding with probability at least $1-\delta$. Fix $d_{\text{VC}} = 50$ and $\delta = 0.05$ and plot these bounds as a function of N. Which bound is the smallest for very large N, say $N = 10,000$? Note that **[c]** and **[d]** are implicit bounds in ϵ.

[a] Original VC bound: $\epsilon \leq \sqrt{\frac{8}{N} \ln \frac{4m_{\mathcal{H}}(2N)}{\delta}}$

[b] Rademacher Penalty Bound: $\epsilon \leq \sqrt{\frac{2\ln(2Nm_{\mathcal{H}}(N))}{N}} + \sqrt{\frac{2}{N} \ln \frac{1}{\delta}} + \frac{1}{N}$

[c] Parrondo and Van den Broek: $\epsilon \leq \sqrt{\frac{1}{N}(2\epsilon + \ln \frac{6m_{\mathcal{H}}(2N)}{\delta})}$

[d] Devroye: $\epsilon \leq \sqrt{\frac{1}{2N}(4\epsilon(1+\epsilon) + \ln \frac{4m_{\mathcal{H}}(N^2)}{\delta})}$

[e] They are all equal.

3. For the same values of d_{VC} and δ of Problem 2, but for small N, say $N = 5$, which bound is the smallest?

[a] Original VC bound: $\epsilon \leq \sqrt{\frac{8}{N} \ln \frac{4m_{\mathcal{H}}(2N)}{\delta}}$

[b] Rademacher Penalty Bound: $\epsilon \leq \sqrt{\frac{2\ln(2Nm_{\mathcal{H}}(N))}{N}} + \sqrt{\frac{2}{N} \ln \frac{1}{\delta}} + \frac{1}{N}$

[c] Parrondo and Van den Broek: $\epsilon \leq \sqrt{\frac{1}{N}(2\epsilon + \ln \frac{6m_{\mathcal{H}}(2N)}{\delta})}$

[d] Devroye: $\epsilon \leq \sqrt{\frac{1}{2N}(4\epsilon(1+\epsilon) + \ln \frac{4m_{\mathcal{H}}(N^2)}{\delta})}$

[e] They are all equal.

• Bias and Variance

Consider the case where the target function $f : [-1, 1] \to \mathbb{R}$ is given by $f(x) = \sin(\pi x)$ and the input probability distribution is uniform on $[-1, 1]$. Assume that the training set has only two examples (picked independently), and that the learning algorithm produces the hypothesis that minimizes the mean squared error on the examples.

4. Assume the learning model consists of all hypotheses of the form $h(x) = ax$. What is the expected value, $\bar{g}(x)$, of the hypothesis produced by the learning algorithm (expected value with respect to the data set)? Express your $\bar{g}(x)$ as $\hat{a}x$, and round $\hat{a}$ to two decimal digits only, then match *exactly* to one of the following answers.

[a] $\bar{g}(x) = 0$

[b] $\bar{g}(x) = 0.79x$

[c] $\bar{g}(x) = 1.07x$

[d] $\bar{g}(x) = 1.58x$

[e] None of the above

5. What is the closest value to the bias in this case?

[a] 0.1

[b] 0.3

[c] 0.5

[d] 0.7

[e] 1.0

6. What is the closest value to the variance in this case?

[a] 0.2

[b] 0.4

[c] 0.6

[d] 0.8

[e] 1.0

7. Now, let's change $\mathcal{H}$. Which of the following learning models has the least expected value of out-of-sample error?

[a] Hypotheses of the form $h(x) = b$

[b] Hypotheses of the form $h(x) = ax$

[c] Hypotheses of the form $h(x) = ax + b$

[d] Hypotheses of the form $h(x) = ax^2$

[e] Hypotheses of the form $h(x) = ax^2 + b$

- **VC Dimension**

8. Assume $q \geq 1$ is an integer and let $m_{\mathcal{H}}(1) = 2$. What is the VC dimension of a hypothesis set whose growth function satisfies: $m_{\mathcal{H}}(N+1) = 2m_{\mathcal{H}}(N) - \binom{N}{q}$? Recall that $\binom{M}{m} = 0$ when $m > M$.

[a] $q - 2$

[b] $q - 1$

[c] q

[d] $q + 1$

[e] None of the above

9. For hypothesis sets $\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_K$ with finite, positive VC dimensions $d_{\text{VC}}(\mathcal{H}_k)$, some of the following bounds are correct and some are not. Which among the correct ones is the tightest bound (the smallest range of values) on the VC dimension of the **intersection** of the sets: $d_{\text{VC}}(\bigcap_{k=1}^{K} \mathcal{H}_k)$? (The VC dimension of an empty set or a singleton set is taken as zero)

[a] $0 \leq d_{\text{VC}}(\bigcap_{k=1}^{K} \mathcal{H}_k) \leq \sum_{k=1}^{K} d_{\text{VC}}(\mathcal{H}_k)$

[b] $0 \leq d_{\text{VC}}(\bigcap_{k=1}^{K} \mathcal{H}_k) \leq \min\{d_{\text{VC}}(\mathcal{H}_k)\}_{k=1}^{K}$

[c] $0 \leq d_{\text{VC}}(\bigcap_{k=1}^{K} \mathcal{H}_k) \leq \max\{d_{\text{VC}}(\mathcal{H}_k)\}_{k=1}^{K}$

[d] $\min\{d_{\text{VC}}(\mathcal{H}_k)\}_{k=1}^{K} \leq d_{\text{VC}}(\bigcap_{k=1}^{K} \mathcal{H}_k) \leq \max\{d_{\text{VC}}(\mathcal{H}_k)\}_{k=1}^{K}$

[e] $\min\{d_{\text{VC}}(\mathcal{H}_k)\}_{k=1}^{K} \leq d_{\text{VC}}(\bigcap_{k=1}^{K} \mathcal{H}_k) \leq \sum_{k=1}^{K} d_{\text{VC}}(\mathcal{H}_k)$

10. For hypothesis sets $\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_K$ with finite, positive VC dimensions $d_{\text{VC}}(\mathcal{H}_k)$, some of the following bounds are correct and some are not. Which among the correct ones is the tightest bound (the smallest range of values) on the VC dimension of the **union** of the sets: $d_{\text{VC}}(\bigcup_{k=1}^{K} \mathcal{H}_k)$?

[a] $0 \leq d_{\text{VC}}(\bigcup_{k=1}^{K} \mathcal{H}_k) \leq \sum_{k=1}^{K} d_{\text{VC}}(\mathcal{H}_k)$

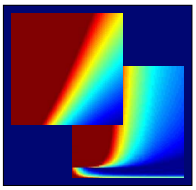
[b] $0 \leq d_{\text{VC}}(\bigcup_{k=1}^{K} \mathcal{H}_k) \leq K - 1 + \sum_{k=1}^{K} d_{\text{VC}}(\mathcal{H}_k)$

[c] $\min\{d_{\mathrm{VC}}(\mathcal{H}_k)\}_{k=1}^{K} \;\leq\; d_{\mathrm{VC}}(\bigcup_{k=1}^{K}\mathcal{H}_k) \;\leq\; \sum_{k=1}^{K} d_{\mathrm{VC}}(\mathcal{H}_k)$

[d] $\max\{d_{\mathrm{VC}}(\mathcal{H}_k)\}_{k=1}^{K} \;\leq\; d_{\mathrm{VC}}(\bigcup_{k=1}^{K}\mathcal{H}_k) \;\leq\; \sum_{k=1}^{K} d_{\mathrm{VC}}(\mathcal{H}_k)$

[e] $\max\{d_{\mathrm{VC}}(\mathcal{H}_k)\}_{k=1}^{K} \;\leq\; d_{\mathrm{VC}}(\bigcup_{k=1}^{K}\mathcal{H}_k) \;\leq\; K-1+\sum_{k=1}^{K} d_{\mathrm{VC}}(\mathcal{H}_k)$

Learning From Data
Yaser Abu-Mostafa, *Caltech*
http://work.caltech.edu/telecourse
Self-paced version



Homework # 5

*All questions have multiple-choice answers (***[a], [b], [c], ...***). You can collaborate with others, but do not discuss the selected or excluded choices in the answers. You can consult books and notes, but not other people's solutions. Your solutions should be based on your own work. Definitions and notation follow the lectures.*

Note about the homework

- The goal of the homework is to facilitate a deeper understanding of the course material. The questions are not designed to be puzzles with catchy answers. They are meant to make you roll up your sleeves, face uncertainties, and approach the problem from different angles.
- The problems range from easy to difficult, and from practical to theoretical. Some problems require running a full experiment to arrive at the answer.
- The answer may not be obvious or numerically close to one of the choices, but one (and only one) choice will be correct if you follow the instructions precisely in each problem. You are encouraged to explore the problem further by experimenting with variations on these instructions, for the learning benefit.
- You are also encouraged to take part in the forum

  http://book.caltech.edu/bookforum

  where there are many threads about each homework set. We hope that you will contribute to the discussion as well. Please follow the forum guidelines for posting answers (see the "BEFORE posting answers" announcement at the top there).

• Linear Regression Error

Consider a noisy target $y = \mathbf{w}^{*T}\mathbf{x} + \epsilon$, where $\mathbf{x} \in \mathbb{R}^d$ (with the added coordinate $x_0 = 1$), $y \in \mathbb{R}$, $\mathbf{w}^*$ is an unknown vector, and ϵ is a noise term with zero mean and σ^2 variance. Assume ϵ is independent of $\mathbf{x}$ and of all other ϵ's. If linear regression is carried out using a training data set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, and outputs the parameter vector $\mathbf{w}_{\text{lin}}$, it can be shown that the expected in-sample error E_{in} with respect to $\mathcal{D}$ is given by:

$$\mathbb{E}_{\mathcal{D}}[E_{\text{in}}(\mathbf{w}_{\text{lin}})] = \sigma^2 \left(1 - \frac{d+1}{N}\right)$$

1. For $\sigma = 0.1$ and $d = 8$, which among the following choices is the smallest number of examples N that will result in an expected E_{in} greater than 0.008?

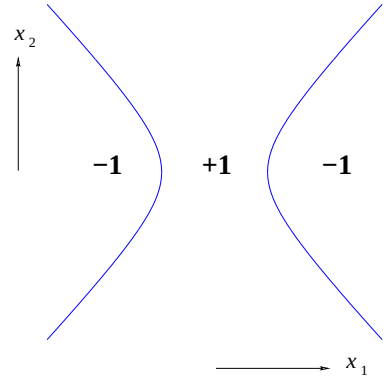
[a] 10
[b] 25
[c] 100
[d] 500
[e] 1000

• Nonlinear Transforms

In linear classification, consider the feature transform $\Phi : \mathbb{R}^2 \to \mathbb{R}^2$ (plus the added zeroth coordinate) given by:

$$\Phi(1, x_1, x_2) = (1, x_1^2, x_2^2)$$

2. Which of the following sets of constraints on the weights in the $\mathcal{Z}$ space could correspond to the hyperbolic decision boundary in $\mathcal{X}$ depicted in the figure?



You may assume that $\tilde{w}_0$ can be selected to achieve the desired boundary.

[a] $\tilde{w}_1 = 0, \tilde{w}_2 > 0$

[b] $\tilde{w}_1 > 0, \tilde{w}_2 = 0$

[c] $\tilde{w}_1 > 0, \tilde{w}_2 > 0$

[d] $\tilde{w}_1 < 0, \tilde{w}_2 > 0$

[e] $\tilde{w}_1 > 0, \tilde{w}_2 < 0$

Now, consider the 4th order polynomial transform from the input space $\mathbb{R}^2$:

$$\Phi_4 : \mathbf{x} \rightarrow (1, x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3, x_1^4, x_1^3x_2, x_1^2x_2^2, x_1x_2^3, x_2^4)$$

3. What is the smallest value among the following choices that is *not* smaller than the VC dimension of a linear model in this transformed space?

[a] 3

[b] 5

[c] 15

[d] 20

[e] 21

• Gradient Descent

Consider the nonlinear error surface $E(u, v) = (ue^v - 2ve^{-u})^2$. We start at the point $(u, v) = (1, 1)$ and minimize this error using gradient descent in the uv space. Use $\eta = 0.1$ (learning rate, not step size).

4. What is the partial derivative of $E(u, v)$ with respect to u, i.e., $\frac{\partial E}{\partial u}$?

[a] $(ue^v - 2ve^{-u})^2$

[b] $2(ue^v - 2ve^{-u})$

[c] $2(e^v + 2ve^{-u})$

[d] $2(e^v - 2ve^{-u})(ue^v - 2ve^{-u})$

[e] $2(e^v + 2ve^{-u})(ue^v - 2ve^{-u})$

5. How many iterations (among the given choices) does it take for the error $E(u, v)$ to fall below 10^{-14} for the first time? In your programs, make sure to use double precision to get the needed accuracy.

[a] 1

[**b**] 3

[**c**] 5

[**d**] 10

[**e**] 17

6. After running enough iterations such that the error has just dropped below 10^{-14}, what are the closest values (in Euclidean distance) among the following choices to the final (u, v) you got in Problem 5?

[**a**] $(1.000, 1.000)$

[**b**] $(0.713, 0.045)$

[**c**] $(0.016, 0.112)$

[**d**] $(-0.083, 0.029)$

[**e**] $(0.045, 0.024)$

7. Now, we will compare the performance of "coordinate descent." In each iteration, we have two steps along the 2 coordinates. Step 1 is to move only along the u coordinate to reduce the error (assume first-order approximation holds like in gradient descent), and step 2 is to reevaluate and move only along the v coordinate to reduce the error (again, assume first-order approximation holds). Use the same learning rate of $\eta = 0.1$ as we did in gradient descent. What will the error $E(u, v)$ be closest to after 15 full iterations (30 steps)?

[**a**] 10^{-1}

[**b**] 10^{-7}

[**c**] 10^{-14}

[**d**] 10^{-17}

[**e**] 10^{-20}

• Logistic Regression

In this problem you will create your own target function f (probability in this case) and data set $\mathcal{D}$ to see how Logistic Regression works. For simplicity, we will take f to be a 0/1 probability so y is a deterministic function of $\mathbf{x}$.

Take $d = 2$ so you can visualize the problem, and let $\mathcal{X} = [-1, 1] \times [-1, 1]$ with uniform probability of picking each $\mathbf{x} \in \mathcal{X}$. Choose a line in the plane as the boundary between $f(\mathbf{x}) = 1$ (where y has to be $+1$) and $f(\mathbf{x}) = 0$ (where y has to be -1) by taking two random, uniformly distributed points from $\mathcal{X}$ and taking the line passing through

them as the boundary between $y = \pm 1$. Pick $N = 100$ training points at random from $\mathcal{X}$, and evaluate the outputs y_n for each of these points $\mathbf{x}_n$.

Run Logistic Regression with Stochastic Gradient Descent to find g, and estimate E_{out} (the **cross entropy** error) by generating a sufficiently large, separate set of points to evaluate the error. Repeat the experiment for 100 runs with different targets and take the average. Initialize the weight vector of Logistic Regression to all zeros in each run. Stop the algorithm when $\|\mathbf{w}^{(t-1)} - \mathbf{w}^{(t)}\| < 0.01$, where $\mathbf{w}^{(t)}$ denotes the weight vector at the end of epoch t. An epoch is a full pass through the N data points (use a random permutation of $1, 2, \cdots, N$ to present the data points to the algorithm within each epoch, and use different permutations for different epochs). Use a learning rate of 0.01.

8. Which of the following is closest to E_{out} for $N = 100$?

[a] 0.025
[b] 0.050
[c] 0.075
[d] 0.100
[e] 0.125

9. How many epochs does it take on average for Logistic Regression to converge for $N = 100$ using the above initialization and termination rules and the specified learning rate? Pick the value that is closest to your results.

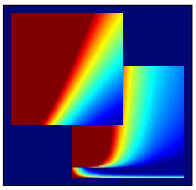
[a] 350
[b] 550
[c] 750
[d] 950
[e] 1750

- **PLA as SGD**

10. The Perceptron Learning Algorithm can be implemented as SGD using which of the following error functions $\mathrm{e}_n(\mathbf{w})$? Ignore the points $\mathbf{w}$ at which $\mathrm{e}_n(\mathbf{w})$ is not twice differentiable.

[a] $\mathrm{e}_n(\mathbf{w}) = e^{-y_n \mathbf{w}^\intercal \mathbf{x}_n}$
[b] $\mathrm{e}_n(\mathbf{w}) = -y_n \mathbf{w}^\intercal \mathbf{x}_n$
[c] $\mathrm{e}_n(\mathbf{w}) = (y_n - \mathbf{w}^\intercal \mathbf{x}_n)^2$
[d] $\mathrm{e}_n(\mathbf{w}) = \ln(1 + e^{-y_n \mathbf{w}^\intercal \mathbf{x}_n})$
[e] $\mathrm{e}_n(\mathbf{w}) = -\min(0, y_n \mathbf{w}^\intercal \mathbf{x}_n)$

Learning From Data
Yaser Abu-Mostafa, *Caltech*
http://work.caltech.edu/telecourse
Self-paced version



Homework # 6

*All questions have multiple-choice answers (***[a], [b], [c], ...***). You can collaborate with others, but do not discuss the selected or excluded choices in the answers. You can consult books and notes, but not other people's solutions. Your solutions should be based on your own work. Definitions and notation follow the lectures.*

Note about the homework

- The goal of the homework is to facilitate a deeper understanding of the course material. The questions are not designed to be puzzles with catchy answers. They are meant to make you roll up your sleeves, face uncertainties, and approach the problem from different angles.
- The problems range from easy to difficult, and from practical to theoretical. Some problems require running a full experiment to arrive at the answer.
- The answer may not be obvious or numerically close to one of the choices, but one (and only one) choice will be correct if you follow the instructions precisely in each problem. You are encouraged to explore the problem further by experimenting with variations on these instructions, for the learning benefit.
- You are also encouraged to take part in the forum

  http://book.caltech.edu/bookforum

  where there are many threads about each homework set. We hope that you will contribute to the discussion as well. Please follow the forum guidelines for posting answers (see the "BEFORE posting answers" announcement at the top there).

• Overfitting and Deterministic Noise

1. Deterministic noise depends on $\mathcal{H}$, as some models approximate f better than others. Assume $\mathcal{H}' \subset \mathcal{H}$ and that f is fixed. **In general** (but not necessarily in all cases), if we use $\mathcal{H}'$ instead of $\mathcal{H}$, how does deterministic noise behave?

[**a**] In general, deterministic noise will decrease.

[**b**] In general, deterministic noise will increase.

[**c**] In general, deterministic noise will be the same.

[**d**] There is deterministic noise for only one of $\mathcal{H}$ and $\mathcal{H}'$.

• Regularization with Weight Decay

In the following problems use the data provided in the files

http://work.caltech.edu/data/in.dta

http://work.caltech.edu/data/out.dta

as a training and test set respectively. Each line of the files corresponds to a two-dimensional input $\mathbf{x} = (x_1, x_2)$, so that $\mathcal{X} = \mathbb{R}^2$, followed by the corresponding label from $\mathcal{Y} = \{-1, 1\}$. We are going to apply Linear Regression with a non-linear transformation for classification. The nonlinear transformation is given by

$$\phi(x_1, x_2) = (1, x_1, x_2, x_1^2, x_2^2, x_1 x_2, |x_1 - x_2|, |x_1 + x_2|).$$

Recall that the classification error is defined as the fraction of misclassified points.

2. Run Linear Regression on the training set after performing the non-linear transformation. What values are closest (in Euclidean distance) to the in-sample and out-of-sample classification errors, respectively?

[**a**] 0.03, 0.08

[**b**] 0.03, 0.10

[**c**] 0.04, 0.09

[**d**] 0.04, 0.11

[**e**] 0.05, 0.10

3. Now add weight decay to Linear Regression, that is, add the term $\frac{\lambda}{N} \sum_{i=0}^{7} w_i^2$ to the squared in-sample error, using $\lambda = 10^k$. What are the closest values to the in-sample and out-of-sample classification errors, respectively, for $k = -3$? Recall that the solution for Linear Regression with Weight Decay was derived in class.

[a] 0.01, 0.02

[b] 0.02, 0.04

[c] 0.02, 0.06

[d] 0.03, 0.08

[e] 0.03, 0.10

4. Now, use $k = 3$. What are the closest values to the new in-sample and out-of-sample classification errors, respectively?

[a] 0.2, 0.2

[b] 0.2, 0.3

[c] 0.3, 0.3

[d] 0.3, 0.4

[e] 0.4, 0.4

5. What value of k, among the following choices, achieves the smallest out-of-sample classification error?

[a] 2

[b] 1

[c] 0

[d] -1

[e] -2

6. What value is closest to the minimum out-of-sample classification error achieved by varying k (limiting k to integer values)?

[a] 0.04

[b] 0.06

[c] 0.08

[d] 0.10

[e] 0.12

• Regularization for Polynomials

Polynomial models can be viewed as linear models in a space $\mathcal{Z}$, under a nonlinear transform $\Phi : \mathcal{X} \to \mathcal{Z}$. Here, Φ transforms the scalar x into a vector $\mathbf{z}$ of Legendre

polynomials, $\mathbf{z} = (1, L_1(x), L_2(x), ..., L_Q(x))$. Our hypothesis set will be expressed as a linear combination of these polynomials,

$$\mathcal{H}_Q = \left\{ h \mid h(x) = \mathbf{w}^{\mathrm{T}}\mathbf{z} = \sum_{q=0}^{Q} w_q L_q(x) \right\},$$

where $L_0(x) = 1$.

7. Consider the following hypothesis set defined by the constraint:

$$\mathcal{H}(Q, C, Q_o) = \{h \mid h(x) = \mathbf{w}^{\mathrm{T}}\mathbf{z} \in \mathcal{H}_Q; w_q = C \text{ for } q \geq Q_o\},$$

which of the following statements is correct:

[a] $\mathcal{H}(10, 0, 3) \cup \mathcal{H}(10, 0, 4) = \mathcal{H}_4$

[b] $\mathcal{H}(10, 1, 3) \cup \mathcal{H}(10, 1, 4) = \mathcal{H}_3$

[c] $\mathcal{H}(10, 0, 3) \cap \mathcal{H}(10, 0, 4) = \mathcal{H}_2$

[d] $\mathcal{H}(10, 1, 3) \cap \mathcal{H}(10, 1, 4) = \mathcal{H}_1$

[e] None of the above

- **Neural Networks**

8. A fully connected Neural Network has $L = 2$; $d^{(0)} = 5$, $d^{(1)} = 3$, $d^{(2)} = 1$. If only products of the form $w_{ij}^{(l)} x_i^{(l-1)}$, $w_{ij}^{(l)} \delta_j^{(l)}$, and $x_i^{(l-1)} \delta_j^{(l)}$ count as operations (even for $x_0^{(l-1)} = 1$), without counting anything else, which of the following is the closest to the total number of operations in a single iteration of backpropagation (using SGD on one data point)?

[a] 30

[b] 35

[c] 40

[d] 45

[e] 50

Let us call every 'node' in a Neural Network a unit, whether that unit is an input variable or a neuron in one of the layers. Consider a Neural Network that has 10 input units (the constant $x_0^{(0)}$ is counted here as a unit), one output unit, and 36 hidden units (each $x_0^{(l)}$ is also counted as a unit). The hidden units can be arranged in any number of layers $l = 1, \cdots, L-1$, and each layer is fully connected to the layer above it.

9. What is the minimum possible number of weights that such a network can have?

[a] 46

[b] 47

[c] 56

[d] 57

[e] 58

10. What is the maximum possible number of weights that such a network can have?

[a] 386

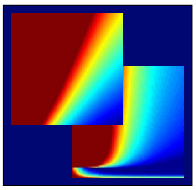
[b] 493

[c] 494

[d] 509

[e] 510

Learning From Data
Yaser Abu-Mostafa, *Caltech*
http://work.caltech.edu/telecourse
Self-paced version



Homework # 7

*All questions have multiple-choice answers (***[a], [b], [c], ...***). You can collaborate with others, but do not discuss the selected or excluded choices in the answers. You can consult books and notes, but not other people's solutions. Your solutions should be based on your own work. Definitions and notation follow the lectures.*

Note about the homework

- The goal of the homework is to facilitate a deeper understanding of the course material. The questions are not designed to be puzzles with catchy answers. They are meant to make you roll up your sleeves, face uncertainties, and approach the problem from different angles.
- The problems range from easy to difficult, and from practical to theoretical. Some problems require running a full experiment to arrive at the answer.
- The answer may not be obvious or numerically close to one of the choices, but one (and only one) choice will be correct if you follow the instructions precisely in each problem. You are encouraged to explore the problem further by experimenting with variations on these instructions, for the learning benefit.
- You are also encouraged to take part in the forum

  http://book.caltech.edu/bookforum

  where there are many threads about each homework set. We hope that you will contribute to the discussion as well. Please follow the forum guidelines for posting answers (see the "BEFORE posting answers" announcement at the top there).

- **Validation**

In the following problems, use the data provided in the files `in.dta` and `out.dta` for Homework # 6. We are going to apply linear regression with a nonlinear transformation for classification (without regularization). The nonlinear transformation is given by ϕ_0 through ϕ_7 which transform (x_1, x_2) into

$$1 \qquad x_1 \qquad x_2 \qquad x_1^2 \qquad x_2^2 \qquad x_1 x_2 \qquad |x_1 - x_2| \qquad |x_1 + x_2|$$

To illustrate how taking out points for validation affects the performance, we will consider the hypotheses trained on $\mathcal{D}_{\text{train}}$ (without restoring the full $\mathcal{D}$ for training after validation is done).

1. Split `in.dta` into training (first 25 examples) and validation (last 10 examples). Train on the 25 examples only, using the validation set of 10 examples to select between five models that apply linear regression to ϕ_0 through ϕ_k, with $k = 3, 4, 5, 6, 7$. For which model is the classification error on the validation set smallest?

[**a**] $k = 3$
[**b**] $k = 4$
[**c**] $k = 5$
[**d**] $k = 6$
[**e**] $k = 7$

2. Evaluate the out-of-sample classification error using `out.dta` on the 5 models to see how well the validation set predicted the best of the 5 models. For which model is the out-of-sample classification error smallest?

[**a**] $k = 3$
[**b**] $k = 4$
[**c**] $k = 5$
[**d**] $k = 6$
[**e**] $k = 7$

3. Reverse the role of training and validation sets; now training with the last 10 examples and validating with the first 25 examples. For which model is the classification error on the validation set smallest?

[**a**] $k = 3$
[**b**] $k = 4$

[**c**] $k = 5$

[**d**] $k = 6$

[**e**] $k = 7$

4. Once again, evaluate the out-of-sample classification error using `out.dta` on the 5 models to see how well the validation set predicted the best of the 5 models. For which model is the out-of-sample classification error smallest?

[**a**] $k = 3$

[**b**] $k = 4$

[**c**] $k = 5$

[**d**] $k = 6$

[**e**] $k = 7$

5. What values are closest in Euclidean distance to the out-of-sample classification error obtained for the model chosen in Problems 1 and 3, respectively?

[**a**] 0.0, 0.1

[**b**] 0.1, 0.2

[**c**] 0.1, 0.3

[**d**] 0.2, 0.2

[**e**] 0.2, 0.3

- **Validation Bias**

6. Let e_1 and e_2 be independent random variables, distributed uniformly over the interval $[0, 1]$. Let $\mathsf{e} = \min(\mathsf{e}_1, \mathsf{e}_2)$. The expected values of $\mathsf{e}_1, \mathsf{e}_2, \mathsf{e}$ are closest to

[**a**] 0.5, 0.5, 0

[**b**] 0.5, 0.5, 0.1

[**c**] 0.5, 0.5, 0.25

[**d**] 0.5, 0.5, 0.4

[**e**] 0.5, 0.5, 0.5

- **Cross Validation**

7. You are given the data points (x, y): $(-1, 0), (\rho, 1), (1, 0)$, $\rho \geq 0$, and a choice between two models: constant $\{\, h_0(x) = b \,\}$ and linear $\{\, h_1(x) = ax + b \,\}$. For which value of ρ would the two models be tied using leave-one-out cross-validation with the squared error measure?

[a] $\sqrt{\sqrt{3}+4}$

[b] $\sqrt{\sqrt{3}-1}$

[c] $\sqrt{9+4\sqrt{6}}$

[d] $\sqrt{9-\sqrt{6}}$

[e] None of the above

• PLA vs. SVM

Notice: Quadratic Programming packages sometimes need tweaking and have numerical issues, and this is characteristic of packages you will use in practical ML situations. Your understanding of support vectors will help you get to the correct answers.

In the following problems, we compare PLA to SVM with hard margin[1] on linearly separable data sets. For each run, you will create your own target function f and data set $\mathcal{D}$. Take $d = 2$ and choose a random line in the plane as your target function f (do this by taking two random, uniformly distributed points on $[-1, 1] \times [-1, 1]$ and taking the line passing through them), where one side of the line maps to $+1$ and the other maps to -1. Choose the inputs $\mathbf{x}_n$ of the data set as random points in $\mathcal{X} = [-1, 1] \times [-1, 1]$, and evaluate the target function on each $\mathbf{x}_n$ to get the corresponding output y_n. If all data points are on one side of the line, discard the run and start a new run.

Start PLA with the all-zero vector and pick the misclassified point for each PLA iteration at random. Run PLA to find the final hypothesis g_{PLA} and measure the disagreement between f and g_{PLA} as $\mathbb{P}[f(\mathbf{x}) \neq g_{\text{PLA}}(\mathbf{x})]$ (you can either calculate this exactly, or approximate it by generating a sufficiently large, separate set of points to evaluate it). Now, run SVM on the same data to find the final hypothesis g_{SVM} by solving

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \\ \text{s.t.} \quad & y_n\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b\right) \geq 1 \end{aligned}$$

using quadratic programming on the primal or the dual problem. Measure the disagreement between f and g_{SVM} as $\mathbb{P}[f(\mathbf{x}) \neq g_{\text{SVM}}(\mathbf{x})]$, and count the number of support vectors you get in each run.

8. For $N = 10$, repeat the above experiment for 1000 runs. How often is g_{SVM} better than g_{PLA} in approximating f? The percentage of time is closest to:

[a] 20%

[1] For hard margin in SVM packages, set $C \to \infty$.

[**b**] 40%

[**c**] 60%

[**d**] 80%

[**e**] 100%

9. For $N = 100$, repeat the above experiment for 1000 runs. How often is g_{SVM} better than g_{PLA} in approximating f? The percentage of time is closest to:

[**a**] 10%

[**b**] 30%

[**c**] 50%

[**d**] 70%

[**e**] 90%

10. For the case $N = 100$, which of the following is the closest to the average number of support vectors of g_{SVM} (averaged over the 1000 runs)?

[**a**] 2

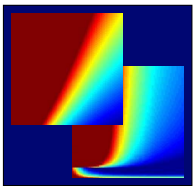
[**b**] 3

[**c**] 5

[**d**] 10

[**e**] 20

Learning From Data
Yaser Abu-Mostafa, *Caltech*
http://work.caltech.edu/telecourse
Self-paced version



Homework # 8

*All questions have multiple-choice answers (***[a], [b], [c], ...***). You can collaborate with others, but do not discuss the selected or excluded choices in the answers. You can consult books and notes, but not other people's solutions. Your solutions should be based on your own work. Definitions and notation follow the lectures.*

Note about the homework

- The goal of the homework is to facilitate a deeper understanding of the course material. The questions are not designed to be puzzles with catchy answers. They are meant to make you roll up your sleeves, face uncertainties, and approach the problem from different angles.
- The problems range from easy to difficult, and from practical to theoretical. Some problems require running a full experiment to arrive at the answer.
- The answer may not be obvious or numerically close to one of the choices, but one (and only one) choice will be correct if you follow the instructions precisely in each problem. You are encouraged to explore the problem further by experimenting with variations on these instructions, for the learning benefit.
- You are also encouraged to take part in the forum

  http://book.caltech.edu/bookforum

  where there are many threads about each homework set. We hope that you will contribute to the discussion as well. Please follow the forum guidelines for posting answers (see the "BEFORE posting answers" announcement at the top there).

- **Primal versus Dual Problem**

1. Recall that N is the size of the data set and d is the dimensionality of the input space. The original formulation of the hard-margin SVM problem (minimize $\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$ subject to the inequality constraints), without going through the Lagrangian dual problem, is

[a] a quadratic programming problem with N variables

[b] a quadratic programming problem with $N + 1$ variables

[c] a quadratic programming problem with d variables

[d] a quadratic programming problem with $d + 1$ variables

[e] not a quadratic programming problem

Notice: The following problems deal with a real-life data set. In addition, the computational packages you use may employ different heuristics and require different tweaks. This is a typical situation that a Machine Learning practitioner faces. There are uncertainties, and the answers may or may not match our expectations. Although this situation is not as 'sanitized' as other homework problems, it is important to go through it as part of the learning experience.

SVM with Soft Margins

In the rest of the problems of this homework set, we apply soft-margin SVM to handwritten digits from the processed US Postal Service Zip Code data set. Download the data (extracted features of intensity and symmetry) for training and testing:

http://www.amlbook.com/data/zip/features.train

http://www.amlbook.com/data/zip/features.test

(the format of each row is: **digit intensity symmetry**). We will train two types of binary classifiers; one-versus-one (one digit is class $+1$ and another digit is class -1, with the rest of the digits disregarded), and one-versus-all (one digit is class $+1$ and the rest of the digits are class -1).

The data set has thousands of points, and some quadratic programming packages cannot handle this size. We recommend that you use the packages in libsvm:

http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Implement SVM with soft margin on the above zip-code data set by solving

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\alpha_n\alpha_m y_n y_m K(\mathbf{x}_n,\mathbf{x}_m) - \sum_{n=1}^{N}\alpha_n \\ \text{s.t.} \quad & \sum_{n=1}^{N} y_n\alpha_n = 0 \\ & 0 \le \alpha_n \le C \quad n = 1,\cdots,N \end{aligned}$$

When evaluating E_{in} and E_{out} of the resulting classifier, use binary classification error.

Practical remarks:

(i) For the purpose of this homework, do not scale the data when you use libsvm or other packages, otherwise you may inadvertently change the (effective) kernel and get different results.

(ii) In some packages, you need to specify double precision.

(iii) In 10-fold cross validation, if the data size is not a multiple of 10, the sizes of the 10 subsets may be off by 1 data point.

(iv) Some packages have software parameters whose values affect the outcome. ML practitioners have to deal with this kind of added uncertainty.

- **Polynomial Kernels**

Consider the polynomial kernel $K(\mathbf{x}_n,\mathbf{x}_m) = (1+\mathbf{x}_n^{\mathrm{T}}\mathbf{x}_m)^Q$, where Q is the degree of the polynomial.

2. With $C = 0.01$ and $Q = 2$, which of the following classifiers has the **highest** E_{in}?

[a] 0 versus all

[b] 2 versus all

[c] 4 versus all

[d] 6 versus all

[e] 8 versus all

3. With $C = 0.01$ and $Q = 2$, which of the following classifiers has the **lowest** E_{in}?

[a] 1 versus all

[b] 3 versus all

[c] 5 versus all

[**d**] 7 versus all

[**e**] 9 versus all

4. Comparing the two selected classifiers from Problems 2 and 3, which of the following values is the closest to the difference between the number of support vectors of these two classifiers?

[**a**] 600

[**b**] 1200

[**c**] 1800

[**d**] 2400

[**e**] 3000

5. Consider the 1 versus 5 classifier with $Q = 2$ and $C \in \{0.001, 0.01, 0.1, 1\}$. Which of the following statements is correct? Going up or down means strictly so.

[**a**] The number of support vectors goes down when C goes up.

[**b**] The number of support vectors goes up when C goes up.

[**c**] E_{out} goes down when C goes up.

[**d**] Maximum C achieves the lowest E_{in}.

[**e**] None of the above

6. In the 1 versus 5 classifier, comparing $Q = 2$ with $Q = 5$, which of the following statements is correct?

[**a**] When $C = 0.0001$, E_{in} is higher at $Q = 5$.

[**b**] When $C = 0.001$, the number of support vectors is lower at $Q = 5$.

[**c**] When $C = 0.01$, E_{in} is higher at $Q = 5$.

[**d**] When $C = 1$, E_{out} is lower at $Q = 5$.

[**e**] None of the above

- **Cross Validation**

In the next two problems, we will experiment with 10-fold cross validation for the polynomial kernel. Because E_{cv} is a random variable that depends on the random partition of the data, we will try 100 runs with different partitions and base our answer on how many runs lead to a particular choice.

7. Consider the 1 versus 5 classifier with $Q = 2$. We use E_{cv} to select $C \in \{0.0001, 0.001, 0.01, 0.1, 1\}$. If there is a tie in E_{cv}, select the smaller C. Within the 100 random runs, which of the following statements is correct?

[a] $C = 0.0001$ is selected most often.

[b] $C = 0.001$ is selected most often.

[c] $C = 0.01$ is selected most often.

[d] $C = 0.1$ is selected most often.

[e] $C = 1$ is selected most often.

8. Again, consider the 1 versus 5 classifier with $Q = 2$. For the winning selection in the previous problem, the average value of E_{cv} over the 100 runs is closest to

(a) 0.001

(b) 0.003

(c) 0.005

(d) 0.007

(e) 0.009

- **RBF Kernel**

Consider the radial basis function (RBF) kernel $K(\mathbf{x}_n, \mathbf{x}_m) = \exp\left(-||\mathbf{x}_n - \mathbf{x}_m||^2\right)$ in the soft-margin SVM approach. Focus on the 1 versus 5 classifier.

9. Which of the following values of C results in the lowest E_{in}?

[a] $C = 0.01$

[b] $C = 1$

[c] $C = 100$

[d] $C = 10^4$

[e] $C = 10^6$

10. Which of the following values of C results in the lowest E_{out}?

[a] $C = 0.01$

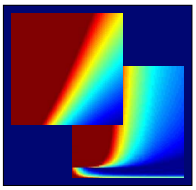
[b] $C = 1$

[c] $C = 100$

[d] $C = 10^4$

[e] $C = 10^6$

Learning From Data
Yaser Abu-Mostafa, *Caltech*
http://work.caltech.edu/telecourse
Self-paced version



Final Exam

All questions have multiple-choice answers ([a], [b], [c], ...*). You can collaborate with others, but do not discuss the selected or excluded choices in the answers. You can consult books and notes, but not other people's solutions. Your solutions should be based on your own work. Definitions and notation follow the lectures.*

Note about the final

- There are twice as many problems in this final as there are in a homework set, and some problems require packages that will need time to get to work properly.
- Problems cover different parts of the course. To facilitate your search for relevant lecture parts, an indexed version of the lecture video segments can be found at the Machine Learning Video Library:

  http://work.caltech.edu/library

- To discuss the final, you are encouraged to take part in the forum

  http://book.caltech.edu/bookforum

  where there is a dedicated subforum for this final.

- Please follow the forum guidelines for posting answers (see the "BEFORE posting answers" announcement at the top there).

- **Nonlinear transforms**

1. The polynomial transform of order $Q = 10$ applied to $\mathcal{X}$ of dimension $d = 2$ results in a $\mathcal{Z}$ space of what dimensionality (not counting the constant coordinate $x_0 = 1$ or $z_0 = 1$)?

[**a**] 12

[**b**] 20

[**c**] 35

[**d**] 100

[**e**] None of the above

- **Bias and Variance**

2. Recall that the average hypothesis $\bar{g}$ was based on training the same model $\mathcal{H}$ on different data sets $\mathcal{D}$ to get $g^{(\mathcal{D})} \in \mathcal{H}$, and taking the expected value of $g^{(\mathcal{D})}$ w.r.t. $\mathcal{D}$ to get $\bar{g}$. Which of the following models $\mathcal{H}$ could result in $\bar{g} \notin \mathcal{H}$?

[**a**] A singleton $\mathcal{H}$ ($\mathcal{H}$ has one hypothesis)

[**b**] $\mathcal{H}$ is the set of all constant, real-valued hypotheses

[**c**] $\mathcal{H}$ is the linear regression model

[**d**] $\mathcal{H}$ is the logistic regression model

[**e**] None of the above

- **Overfitting**

3. Which of the following statements is *false*?

[**a**] If there is overfitting, there must be two or more hypotheses that have different values of E_{in}.

[**b**] If there is overfitting, there must be two or more hypotheses that have different values of E_{out}.

[**c**] If there is overfitting, there must be two or more hypotheses that have different values of $(E_{\text{out}} - E_{\text{in}})$.

[**d**] We can always determine if there is overfitting by comparing the values of $(E_{\text{out}} - E_{\text{in}})$.

[**e**] We cannot determine overfitting based on one hypothesis only.

4. Which of the following statements is true?

[**a**] Deterministic noise cannot occur with stochastic noise.

[**b**] Deterministic noise does not depend on the hypothesis set.

[**c**] Deterministic noise does not depend on the target function.

[**d**] Stochastic noise does not depend on the hypothesis set.

[**e**] Stochastic noise does not depend on the target distribution.

• Regularization

5. The regularized weight $\mathbf{w}_{\text{reg}}$ is a solution to:

$$\text{minimize} \quad \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}^{\text{T}}\mathbf{x}_n - y_n)^2 \quad \text{subject to} \quad \mathbf{w}^{\text{T}}\Gamma^{\text{T}}\Gamma\mathbf{w} \leq C,$$

where Γ is a matrix. If $\mathbf{w}_{\text{lin}}^{\text{T}}\Gamma^{\text{T}}\Gamma\mathbf{w}_{\text{lin}} \leq C$, where $\mathbf{w}_{\text{lin}}$ is the linear regression solution, then what is $\mathbf{w}_{\text{reg}}$?

[**a**] $\mathbf{w}_{\text{reg}} = \mathbf{w}_{\text{lin}}$

[**b**] $\mathbf{w}_{\text{reg}} = \Gamma\mathbf{w}_{\text{lin}}$

[**c**] $\mathbf{w}_{\text{reg}} = \Gamma^{\text{T}}\Gamma\mathbf{w}_{\text{lin}}$

[**d**] $\mathbf{w}_{\text{reg}} = C\Gamma\mathbf{w}_{\text{lin}}$

[**e**] $\mathbf{w}_{\text{reg}} = C\mathbf{w}_{\text{lin}}$

6. Soft-order constraints that regularize polynomial models can be

[**a**] written as hard-order constraints

[**b**] translated into augmented error

[**c**] determined from the value of the VC dimension

[**d**] used to decrease both E_{in} and E_{out}

[**e**] None of the above is true

• Regularized Linear Regression

We are going to experiment with linear regression for classification on the processed US Postal Service Zip Code data set from Homework 8. Download the data (extracted features of intensity and symmetry) for training and testing:

http://www.amlbook.com/data/zip/features.train

http://www.amlbook.com/data/zip/features.test

(the format of each row is: **digit intensity symmetry**). We will train two types of binary classifiers; one-versus-one (one digit is class $+1$ and another digit is class -1, with the rest of the digits disregarded), and one-versus-all (one digit is class $+1$ and the rest of the digits are class -1). When evaluating E_{in} and E_{out}, use binary classification error. Implement the regularized least-squares linear regression for classification that minimizes

$$\frac{1}{N}\sum_{n=1}^{N}\left(\mathbf{w}^{\mathrm{T}}\mathbf{z}_n - y_n\right)^2 \; + \; \frac{\lambda}{N}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

where $\mathbf{w}$ includes w_0.

7. Set $\lambda = 1$ and do not apply a feature transform (i.e., use $\mathbf{z} = \mathbf{x} = (1, x_1, x_2)$). Which among the following classifiers has the lowest E_{in}?

[a] 5 versus all

[b] 6 versus all

[c] 7 versus all

[d] 8 versus all

[e] 9 versus all

8. Now, apply a feature transform $\mathbf{z} = (1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$, and set $\lambda = 1$. Which among the following classifiers has the lowest E_{out}?

[a] 0 versus all

[b] 1 versus all

[c] 2 versus all

[d] 3 versus all

[e] 4 versus all

9. If we compare using the transform versus not using it, and apply that to '0 versus all' through '9 versus all', which of the following statements is correct for $\lambda = 1$?

[a] Overfitting always occurs when we use the transform.

[b] The transform always improves the out-of-sample performance by at least 5% (E_{out} with transform $\leq 0.95 E_{\text{out}}$ without transform).

[c] The transform does not make any difference in the out-of-sample performance.

[d] The transform always worsens the out-of-sample performance by at least 5%.

[e] The transform improves the out-of-sample performance of '5 versus all,' but by less than 5%.

10. Train the '1 versus 5' classifier with $\mathbf{z} = (1, x_1, x_2, x_1x_2, x_1^2, x_2^2)$ with $\lambda = 0.01$ and $\lambda = 1$. Which of the following statements is correct?

[a] Overfitting occurs (from $\lambda = 1$ to $\lambda = 0.01$).

[b] The two classifiers have the same E_{in}.

[c] The two classifiers have the same E_{out}.

[d] When λ goes up, both E_{in} and E_{out} go up.

[e] When λ goes up, both E_{in} and E_{out} go down.

• Support Vector Machines

11. Consider the following training set generated from a target function $f : \mathcal{X} \to \{-1, +1\}$ where $\mathcal{X} = \mathbb{R}^2$

$$\begin{array}{ccc} \mathbf{x}_1 = (1,0), y_1 = -1 & \mathbf{x}_2 = (0,1), y_2 = -1 & \mathbf{x}_3 = (0,-1), y_3 = -1 \\ \mathbf{x}_4 = (-1,0), y_4 = +1 & \mathbf{x}_5 = (0,2), y_5 = +1 & \mathbf{x}_6 = (0,-2), y_6 = +1 \\ & \mathbf{x}_7 = (-2,0), y_7 = +1 & \end{array}$$

Transform this training set into another two-dimensional space $\mathcal{Z}$

$$z_1 = x_2^2 - 2x_1 - 1 \qquad z_2 = x_1^2 - 2x_2 + 1$$

Using geometry (not quadratic programming), what values of $\mathbf{w}$ (without w_0) and b specify the separating plane $\mathbf{w}^{\mathrm{T}}\mathbf{z} + b = 0$ that maximizes the margin in the $\mathcal{Z}$ space? The values of w_1, w_2, b are:

[a] $-1,\ 1,\ -0.5$

[b] $1,\ -1,\ -0.5$

[c] $1,\ 0,\ -0.5$

[d] $0,\ 1,\ -0.5$

[e] None of the above would work.

12. Consider the same training set of the previous problem, but instead of explicitly transforming the input space $\mathcal{X}$, apply the hard-margin SVM algorithm with the kernel

$$K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^{\mathrm{T}}\mathbf{x}')^2$$

(which corresponds to a second-order polynomial transformation). Set up the expression for $\mathcal{L}(\alpha_1 ... \alpha_7)$ and solve for the optimal $\alpha_1, ..., \alpha_7$ (numerically, using a quadratic programming package). The number of support vectors you get is in what range?

[a] 0-1

[b] 2-3

[c] 4-5

[d] 6-7

[e] >7

• Radial Basis Functions

We experiment with the RBF model, both in regular form (Lloyd + pseudo-inverse) with K centers:

$$\text{sign}\left(\sum_{k=1}^{K} w_k \ \exp\left(-\gamma \ ||\mathbf{x} - \mu_k||^2\right) + \ b \right)$$

(notice that there is a bias term), and in kernel form (using the RBF kernel in hard-margin SVM):

$$\text{sign}\left(\sum_{\alpha_n > 0} \alpha_n y_n \ \exp\left(-\gamma \ ||\mathbf{x} - \mathbf{x}_n||^2\right) + \ b \right).$$

The input space is $\mathcal{X} = [-1, 1] \times [-1, 1]$ with uniform probability distribution, and the target is

$$f(\mathbf{x}) = \text{sign}(x_2 - x_1 + 0.25 \sin(\pi x_1))$$

which is slightly nonlinear in the $\mathcal{X}$ space. In each run, generate 100 training points at random using this target, and apply both forms of RBF to these training points. Here are some guidelines:
- Repeat the experiment for as many runs as needed to get the answer to be stable (statistically away from flipping to the closest competing answer).
- In case a data set is not separable in the '$\mathcal{Z}$ space' by the RBF kernel using hard-margin SVM, discard the run but keep track of how often this happens, if ever.

- When you use Lloyd's algorithm, initialize the centers to random points in $\mathcal{X}$ and iterate until there is no change from iteration to iteration. If a cluster becomes empty, discard the run and repeat.

13. For $\gamma = 1.5$, how often do you get a data set that is not separable by the RBF kernel (using hard-margin SVM)? *Hint: Run the hard-margin SVM, then check that the solution has $E_{\text{in}} = 0$.*

[a] $\leq 5\%$ of the time

[b] $> 5\%$ but $\leq 10\%$ of the time

[c] $> 10\%$ but $\leq 20\%$ of the time

[d] $> 20\%$ but $\leq 40\%$ of the time

[e] $> 40\%$ of the time

14. If we use $K = 9$ for regular RBF and take $\gamma = 1.5$, how often does the kernel form beat the regular form (excluding runs mentioned in Problem 13 and runs with empty clusters, if any) in terms of E_{out}?

[a] $\leq 15\%$ of the time

[b] $> 15\%$ but $\leq 30\%$ of the time

[c] $> 30\%$ but $\leq 50\%$ of the time

[d] $> 50\%$ but $\leq 75\%$ of the time

[e] $> 75\%$ of the time

15. If we use $K = 12$ for regular RBF and take $\gamma = 1.5$, how often does the kernel form beat the regular form (excluding runs mentioned in Problem 13 and runs with empty clusters, if any) in terms of E_{out}?

[a] $\leq 10\%$ of the time

[b] $> 10\%$ but $\leq 30\%$ of the time

[c] $> 30\%$ but $\leq 60\%$ of the time

[d] $> 60\%$ but $\leq 90\%$ of the time

[e] $> 90\%$ of the time

16. Now we focus on regular RBF only, with $\gamma = 1.5$. If we go from $K = 9$ clusters to $K = 12$ clusters (only 9 and 12), which of the following 5 cases happens most often in your runs (excluding runs with empty clusters, if any)? Up or down means strictly so.

[a] E_{in} goes down, but E_{out} goes up.

[**b**] E_{in} goes up, but E_{out} goes down.

[**c**] Both E_{in} and E_{out} go up.

[**d**] Both E_{in} and E_{out} go down.

[**e**] E_{in} and E_{out} remain the same.

17. For regular RBF with $K = 9$, if we go from $\gamma = 1.5$ to $\gamma = 2$ (only 1.5 and 2), which of the following 5 cases happens most often in your runs (excluding runs with empty clusters, if any)? Up or down means strictly so.

[**a**] E_{in} goes down, but E_{out} goes up.

[**b**] E_{in} goes up, but E_{out} goes down.

[**c**] Both E_{in} and E_{out} go up.

[**d**] Both E_{in} and E_{out} go down.

[**e**] E_{in} and E_{out} remain the same.

18. What is the percentage of time that regular RBF achieves $E_{\text{in}} = 0$ with $K = 9$ and $\gamma = 1.5$ (excluding runs with empty clusters, if any)?

[**a**] $\leq 10\%$ of the time

[**b**] $> 10\%$ but $\leq 20\%$ of the time

[**c**] $> 20\%$ but $\leq 30\%$ of the time

[**d**] $> 30\%$ but $\leq 50\%$ of the time

[**e**] $> 50\%$ of the time

• Bayesian Priors

19. Let $f \in [0, 1]$ be the unknown probability of getting a heart attack for people in a certain population. Notice that f is just a constant, not a function, for simplicity. We want to model f using a hypothesis $h \in [0, 1]$. Before we see any data, we assume that $P(h = f)$ is uniform over $h \in [0, 1]$ (the prior). We pick one person from the population, and it turns out that he or she had a heart attack. Which of the following is true about the posterior probability that $h = f$ given this sample point?

[**a**] The posterior is uniform over $[0, 1]$.

[**b**] The posterior increases linearly over $[0, 1]$.

[**c**] The posterior increases nonlinearly over $[0, 1]$.

[**d**] The posterior is a delta function at 1 (implying f has to be 1).

[**e**] The posterior cannot be evaluated based on the given information.

- **Aggregation**

20. Given two learned hypotheses g_1 and g_2, we construct the aggregate hypothesis g given by $g(\mathbf{x}) = \frac{1}{2}\left(g_1(\mathbf{x}) + g_2(\mathbf{x})\right)$ for all $\mathbf{x} \in \mathcal{X}$. If we use mean-squared error, which of the following statements is true?

[a] $E_{\text{out}}(g)$ cannot be worse than $E_{\text{out}}(g_1)$.

[b] $E_{\text{out}}(g)$ cannot be worse than the smaller of $E_{\text{out}}(g_1)$ and $E_{\text{out}}(g_2)$.

[c] $E_{\text{out}}(g)$ cannot be worse than the average of $E_{\text{out}}(g_1)$ and $E_{\text{out}}(g_2)$.

[d] $E_{\text{out}}(g)$ has to be between $E_{\text{out}}(g_1)$ and $E_{\text{out}}(g_2)$ (including the end values of that interval).

[e] None of the above