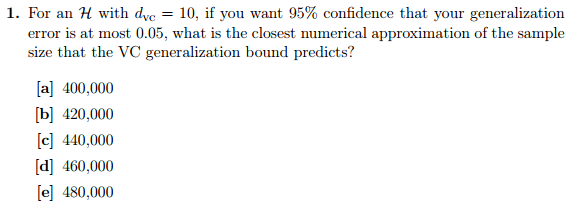
Learning from Data Homework 4



First we shall find a bound on using the general bound formula given VC dimension :

Now we shall work with the VC inequality to get our final equation:

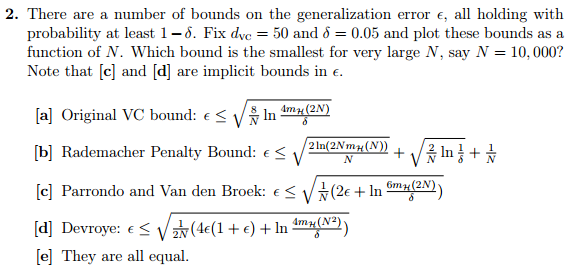
Now we shall enter the available choices to see which is closest to .

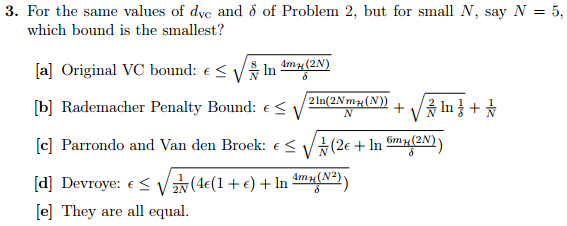
It is obvious that higher sample sizes would depart further from 0.95, so listing those are not necessary. Code implementing this problem can be found within “Learning from Data HW4 Work.R”.

Correction: Didn’t follow instructions. Was instructed to use a simplification of the bound on (shown below) rather than the one above.

Once that was fixed the following answers were achieved:

The closest approximation of the sample size is 440,000.





N delta dVC vc rademacher parrondoV parrondoR devroyeV devroyeR

1: 5 0.10 1 3.0962 2.41063 2.59685 2.22696 6.743648 4.959342

2: 10000 0.10 1 0.1043 0.08339 0.03519 0.03512 0.001238 0.001233

3: 5 0.05 1 3.2704 2.54559 2.70671 2.31305 7.326256 5.350198

4: 10000 0.05 1 0.1069 0.08641 0.03569 0.03562 0.001274 0.001269

5: 5 0.01 1 3.6429 2.80815 2.94385 2.48565 8.666235 6.178443

6: 10000 0.01 1 0.1128 0.09228 0.03682 0.03675 0.001356 0.001351

7: 5 0.10 5 4.0290 2.67451 3.66126 3.03826 13.404853 9.231029

8: 10000 0.10 5 0.1968 0.12272 0.06780 0.06766 0.004597 0.004578

9: 5 0.05 5 4.1644 2.80947 3.73790 3.10651 13.971925 9.650376

10: 10000 0.05 5 0.1982 0.12574 0.06806 0.06792 0.004632 0.004613

11: 5 0.01 5 4.4629 3.07203 3.90926 3.24562 15.282303 10.534050

12: 10000 0.01 5 0.2014 0.13161 0.06866 0.06852 0.004714 0.004695

13: 5 0.10 10 4.0290 2.67451 3.66126 3.03826 13.404853 9.231029

14: 10000 0.10 10 0.2648 0.15339 0.09331 0.09314 0.008707 0.008675

15: 5 0.05 10 4.1644 2.80947 3.73790 3.10651 13.971925 9.650376

16: 10000 0.05 10 0.2658 0.15641 0.09350 0.09333 0.008742 0.008710

17: 5 0.01 10 4.4629 3.07203 3.90926 3.24562 15.282303 10.534050

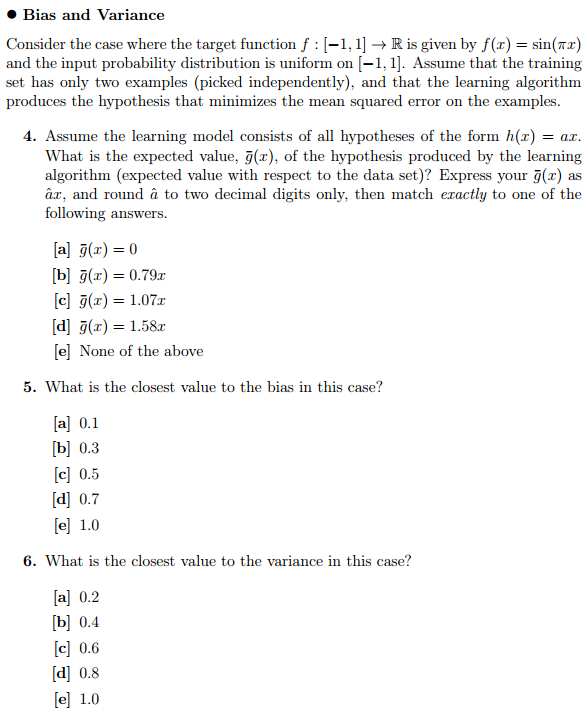
18: 10000 0.01 10 0.2682 0.16228 0.09393 0.09377 0.008823 0.008792

Correction: Same mistake as #1. Once I fixed the approximation, the results were:

|  |
| --- |
| N delta dVC vc rademacher parrondoV parrondoR devroyeV devroyeR  1: 5 0.10 1 3.0962 2.41063 2.40079 1.99486 5.763786 3.979479  2: 10000 0.10 1 0.1043 0.08339 0.03359 0.03352 0.001129 0.001124  3: 5 0.05 1 3.2704 2.54559 2.51920 2.09053 6.346393 4.370335  4: 10000 0.05 1 0.1069 0.08641 0.03411 0.03404 0.001164 0.001159  5: 5 0.01 1 3.6429 2.80815 2.77243 2.28004 7.686373 5.198580  6: 10000 0.01 1 0.1128 0.09228 0.03529 0.03522 0.001246 0.001241  7: 5 0.10 5 4.9318 3.19438 3.69868 2.70882 13.680211 7.337709  8: 10000 0.10 5 0.2063 0.12735 0.06957 0.06941 0.004839 0.004818  9: 5 0.05 5 5.0430 3.32934 3.77328 2.79521 14.237613 7.813186  10: 10000 0.05 5 0.2077 0.13037 0.06982 0.06967 0.004874 0.004854  11: 5 0.01 5 5.2922 3.59190 3.94059 2.96750 15.528260 8.806059  12: 10000 0.01 5 0.2107 0.13624 0.07040 0.07025 0.004956 0.004936  13: 5 0.10 10 6.5379 3.87241 4.82704 3.33691 23.300334 11.134963  14: 10000 0.10 10 0.2867 0.16439 0.09731 0.09712 0.009469 0.009433  15: 5 0.05 10 6.6221 4.00737 4.88334 3.41813 23.846972 11.683646  16: 10000 0.05 10 0.2876 0.16741 0.09749 0.09731 0.009504 0.009469  17: 5 0.01 10 6.8138 4.26994 5.01145 3.58035 25.114619 12.818939  18: 10000 0.01 10 0.2899 0.17328 0.09790 0.09773 0.009585 0.009551 |

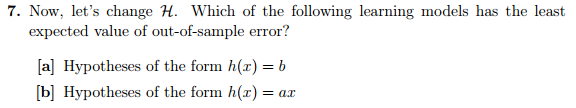
Which makes D the answer for N=10,000 and C the answer for N=5. Averages shown below. Lowest averages are bolded.

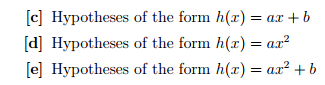
|  |
| --- |
| N vc rademacher parrondoV parrondoR devroyeV devroyeR  5.000 5.028 3.337 3.759 **2.797** 15.056 8.127 |
| N vc rademacher parrondoV parrondoR devroyeV devroyeR  1.000e+04 2.014e-01 1.290e-01 6.728e-02 6.714e-02 5.196e-03 **5.176e-03** |



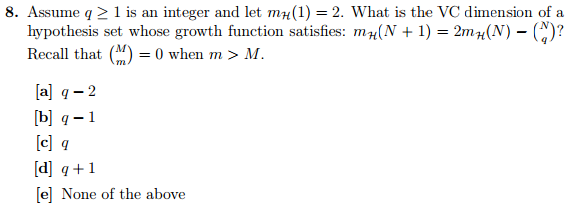
#5 Correction: Accidentally performed in-place operation on real response values throwing off bias (it was 2.0 instead of ~0.25). Real answer is B.

#6 Correction: Using three different pieces of code, consistently got 0.7 (answer should be ). This tells me it’s not a bug, it’s a comprehension error that I’m not seeing (I’ll come back to this later).



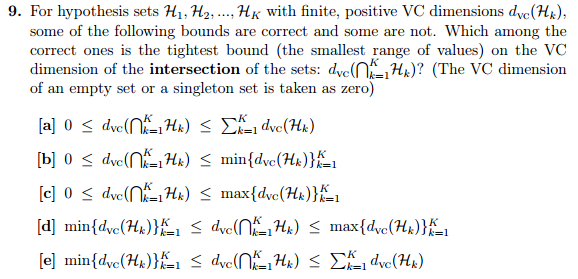


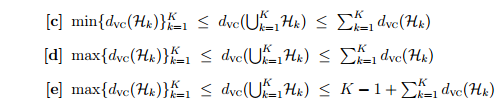
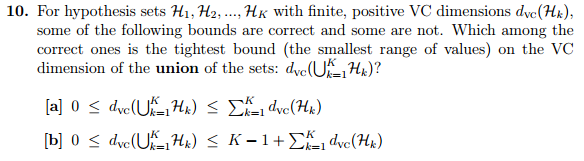
Correction: Given my variance is too high for (b), (a) still holds. I’ll update this to what I should be getting when I figure out why my variance estimates are much too high.



Determined computationally using VCDimen.java inside hw4Programs

Correction: Mixed up definition of VC dimension with definition of break point. Gave the break point of a hypothesis set satisfying that growth function, not it’s VC dimension. Correct answer is C.





Correction: Still don’t quite understand… But the discussions on the forums get an upper bound like e on the basis that , where and . Apparently hardest question in course though, so it’s probably okay to come back to it later.

Homework 4 Solutions

Incorrect problems:

* 1. A, didn’t follow instructions (used more complicated upper bound instead of simplified one)
* 2. B, caused by mistake in #1
* 3. D, caused by mistake in #1
* 5. E, accidentally modified a matrix in-place instead of copying and modifying
* 6. D, consistently getting wrong variance. Will look at later.
* 7. A, caused by mistake in #6
* 8. D, mixed up definition of VC dimension vs break point
* 10. D, misunderstood implications of

2/10. 5/10, if you don’t double count for the same mistake. This was a rough problem set.

