Benjamin Ye
CS/CNS/EE 156a: Learning Systems (Fall 2023)
October 23, 2023

Homework 4

Problem	Answer
1	[d]
2	[d]
3	[c]
4	[e]
5	[b]
6	[a]
7	[b]
8	[c]
9	[b]
10	[e]

Generalization Error

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

1. For an $\mathcal{H}$ with $d_{\mathrm{VC}} = 10$, 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?

   Answer: [d] 460,000

   The VC generalization bound states that for any tolerance $\delta > 0$,

   $$E_{\mathrm{out}}(g) \le E_{\mathrm{in}}(g) + \sqrt{\frac{8}{N}\ln\frac{4m_{\mathcal{H}}(2N)}{\delta}}$$

   with probability $\ge 1-\delta$. With $m_{\mathcal{H}}(N) \le N^{d_{\mathrm{VC}}}$, the bound becomes

   $$E_{\mathrm{out}}(g) \le E_{\mathrm{in}}(g) + \sqrt{\frac{8}{N}\ln\frac{4(2N)^{d_{\mathrm{VC}}}}{\delta}}$$

   Plugging in $d_{\mathrm{VC}} = 10$, $E_{\mathrm{out}}(g) - E_{\mathrm{in}}(g) = 0.05$, and $\delta = 0.05$, and solving for N,

   $$0.05 \le \sqrt{\frac{8}{N}\ln\frac{4(2N)^{10}}{0.05}} \longrightarrow N \ge 452{,}957$$

2. There are a few bounds on the generalization error ϵ, all holding with probability at least $1-\delta$. Fix $d_{\mathrm{VC}} = 50$ 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.

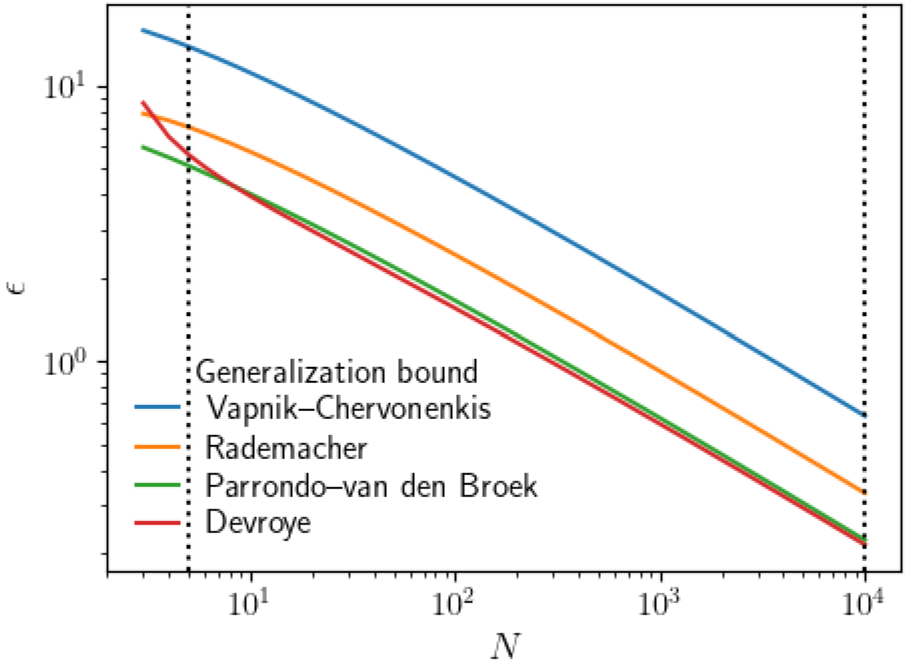
   Answer: [d] Devroye: $\boldsymbol{\epsilon \le \sqrt{\frac{1}{2N}\left(4\epsilon(1+\epsilon) + \ln\frac{4m_{\mathcal{H}}(N^2)}{\delta}\right)}}$

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?

   Answer: [c] Parrondo and van den Broek: $\boldsymbol{\epsilon \le \sqrt{\frac{1}{N}\left(2\epsilon + \ln\frac{6m_{\mathcal{H}}(2N)}{\delta}\right)}}$

See the next page for figures and calculations for problems 2–3.

The figure below shows the relationship between the generalization error bound ϵ as a function of the sample size N, with $N = 5$ and $N = 10{,}000$ highlighted with vertical dotted lines.



The numerical value of the bounds at $N = 10{,}000$ and $N = 5$ are listed below:

```
[HW4 P2-3]
Generalization bounds for d_vc=50 and delta=0.05:
  N=10,000:
    Vapnik-Chervonenkis: 0.632
    Rademacher: 0.331
    Parrondo-van den Broek: 0.224
    Devroye: 0.215
  N=5:
    Vapnik-Chervonenkis: 13.828
    Rademacher: 7.049
    Parrondo-van den Broek: 5.101
    Devroye: 5.593
```

(The Python 3 source code is available on the following page.)

```
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
from scipy import optimize

mpl.rcParams.update(
    {
        "axes.labelsize": 14,
        "figure.autolayout": True,
        "figure.figsize": (4.875, 3.65625),
        "font.size": 12,
        "legend.columnspacing": 1,
        "legend.edgecolor": "1",
        "legend.framealpha": 0,
        "legend.fontsize": 12,
        "legend.handlelength": 1.25,
        "legend.labelspacing": 0.25,
        "xtick.labelsize": 12,
        "ytick.labelsize": 12,
        "text.usetex": True
    }
)

def vapnik_chervonenkis_bound(m_H, N, delta):
    return np.sqrt(8 * np.log(4 * m_H(2 * N) / delta) / N)

def rademacher_bound(m_H, N, delta):
    return (np.sqrt(2 * np.log(2 * N * m_H(N)) / N)
            + np.sqrt(2 * np.log(1 / delta) / N) + 1 / N)

def parrondo_van_den_broek_bound(m_H, N, delta, *, ub=10.0):
    return np.vectorize(
        lambda N: optimize.root_scalar(
            lambda eps: np.sqrt((2 * eps + np.log(6 * m_H(2 * N) / delta)) / N)
                        - eps,
            bracket=(0.0, ub), method="toms748"
        ).root
    )(N)

def devroye_bound(m_H, N, delta, *, ub=10.0, log=False):
    func = lambda eps, N: np.sqrt(
        (4 * eps * (1 + eps) + np.log(4 / delta)
         + (m_H(N ** 2) if log else np.log(m_H(N ** 2)))) / (2 * N)
    ) - eps
    return np.vectorize(
        lambda N: optimize.root_scalar(func, args=N, bracket=(0.0, ub),
                                       method="toms748").root
    )(N)
```

```
if __name__ == "__main__":
    d_vc, delta = 50, 0.05
    m_H = lambda N: N ** d_vc
    Ns = np.arange(3, 10001, dtype=float)
    bounds = {
        "Vapnik-Chervonenkis": vapnik_chervonenkis_bound(m_H, Ns, delta),
        "Rademacher": rademacher_bound(m_H, Ns, delta),
        "Parrondo-van den Broek": parrondo_van_den_broek_bound(m_H, Ns, delta),
        "Devroye": devroye_bound(lambda N: d_vc * np.log(N), Ns, delta, log=True)
    }

    print(f"\n[HW4 P2-3]\nGeneralization bounds for {d_vc=} and {delta=}:")
    for N in (10000, 5):
        i = np.where(Ns == N)[0][0]
        print(f"  {N=:,}:")
        for l, b in bounds.items():
            print(f"    {l}: {b[i]:.3f}")

    _, ax = plt.subplots()
    for l, b in bounds.items():
        ax.plot(Ns, b, label=l)
    ax.set_yscale("log")
    ylim = ax.get_ylim()
    ax.plot((5, 5), ylim, "k:")
    ax.plot((10000, 10000), ylim, "k:")
    ax.legend(title="Generalization bound")
    ax.set_xlabel("$N$")
    ax.set_xscale("log")
    ax.set_ylabel("$\epsilon$")
    ax.set_ylim(ylim)
    ax.text(-0.2, 0.959, " ", transform=ax.transAxes)
    plt.show()
```

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, the match *exactly* to one of the following answers.

   Answer: [e] None of the above

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

   Answer: [b] 0.3

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

   Answer: [a] 0.2

See the next page for derivations, explanations, and simulations for problems 4–6.

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

   Answer: [b] Hypotheses of the form $\boldsymbol{h(x) = ax}$

   For [a] $h(x) = b$, the out-of-sample error is $\mathbb{E}_{\mathcal{D}}[E_{\text{out}}] = \text{bias} + \text{var} = 0.50 + 0.25 = 0.75$ (from example 2.8 in the *Learning from Data* textbook).

   For [b] $h(x) = ax$, the out-of-sample error is $\mathbb{E}_{\mathcal{D}}[E_{\text{out}}] = \text{bias} + \text{var} = 0.270 + 0.236 = 0.506$ (from problem 6).

   For [c] $h(x) = ax + b$, the out-of-sample error is $\mathbb{E}_{\mathcal{D}}[E_{\text{out}}] = \text{bias} + \text{var} = 0.21 + 1.69 = 1.90$ (from example 2.8 in the *Learning from Data* textbook).

   [d] $h(x) = ax^2$ and [e] $h(x) = ax^2 + b$ cannot be used for a data set with a sample size of 2 since they have quadratic forms. Even if it was possible, they are expected to have much higher variances due to their increased complexity compared to hypotheses [a] through [c], and consequently, higher expected out-of-sample errors.

For a sample size of 2, the mean squared error (MSE) of a fit $h(x) = ax$ is given by

$$\text{MSE} = \frac{1}{2}\sum_{i=1}^{2}\left(h(\mathbf{x}_i) - f(\mathbf{x}_i)\right)^2 = \frac{1}{2}\sum_{i=1}^{2}(ax_i - y_i)^2 = \frac{1}{2}[(ax_1 - y_1)^2 + (ax_2 - y_2)^2]$$

By minimizing the MSE with respect to a, we obtain the optimal value of a for the pair of points:

$$\frac{\partial}{\partial a}\text{MSE} = ax_1^2 - x_1y_1 + ax_2^2 - x_2y_2 = 0 \longrightarrow a = \frac{x_1y_1 + x_2y_2}{x_1^2 + x_2^2}$$

To answer problems 4–6, ten million pairs of (x_1, y_1) and (x_2, y_2) are randomly generated, the best fits are determined as outlined above, and $\hat{a}$ is found by averaging the a values from the best fits.

Then, the bias and variance are evaluated on a new set of ten million pairs using

$$\mathbb{E}_\mathbf{x}[\text{bias}(\mathbf{x})] = \frac{1}{N}\sum_{n=1}^{N}\left(\bar{g}(\mathbf{x}_n) - f(\mathbf{x}_n)\right)^2 = \frac{1}{N}\sum_{n=1}^{N}(\hat{a}x_n - \sin(\pi x_n))^2$$

$$\mathbb{E}_\mathbf{x}[\text{var}(\mathbf{x})] = \frac{1}{N}\sum_{n=1}^{N}\left(g^{(\mathcal{D})}(\mathbf{x}_n) - \bar{g}(\mathbf{x}_n)\right)^2 = \frac{1}{N}\sum_{n=1}^{N}(a_nx_n - \hat{a}x_n)^2$$

A sample output is

```
[HW4 P4-6]
Bias and variance of h(x)=ax for f(x)=sin(pi*x):
  g(x)=1.43x, bias=0.270, var=0.236
```

The Python 3 source code is available below.

```
import numpy as np

def generate_data(
        N, f, d=2, lb=-1.0, ub=1.0, *, bias=False, rng=None, seed=None):
    if rng is None:
        rng = np.random.default_rng(seed)
    x = rng.uniform(lb, ub, (N, d))
    if bias:
        x = np.hstack((np.ones((N, 1)), x))
    return x, f(x)

if __name__ == "__main__":
    n_runs = 10_000_000
    x, y = generate_data(2 * n_runs, lambda x: np.sin(np.pi * x), 1)
    as_ = (x[::2] * y[::2] + x[1::2] * y[1::2]) / (x[::2] ** 2 + x[1::2] ** 2)
    a_avg = as_.mean()
    x, y = generate_data(2 * n_runs, lambda x: np.sin(np.pi * x), 1)
    print(f"\n[HW4 P4-6]\nBias and variance of h(x)=ax for f(x)=sin(pi*x):\n"
          f"  g(x)={a_avg:.2f}x, bias={((a_avg * x - y) ** 2).mean():.3f}, "
          f"var={(((np.tile(as_, (2, 1)) - a_avg) * x) ** 2).mean():.3f}")
```

VC Dimension

8. Let $q \geq 1$ be an integer and assume that $m_{\mathcal{H}}(1) = 2$. What is the VC dimension of a hypothesis set whose growth function for all $N \geq 1$ satisfies $m_{\mathcal{H}}(N+1) = 2m_{\mathcal{H}}(N) - C_q^N$? Recall that $C_m^M = 0$ when $m > M$.

   Answer: [c] *q*

   The growth function is

$$m_{\mathcal{H}}(N+1) = 2m_{\mathcal{H}}(N) - \binom{N}{q}$$

   When $N \leq q$, the combination term in the right-hand side of the growth function becomes zero. By starting with $N = 1$ and recursively multiplying by 2 for each increment in N, the growth function simplifies to

$$m_{\mathcal{H}}(N \leq q) = 2^N$$

   Now, let's consider the case when $N > q$. For $N = q + 1$, the growth function is

$$m_{\mathcal{H}}(q+1) = 2m_{\mathcal{H}}(q) - \binom{q}{q} = 2^{q+1} - 1$$

   Since $m_{\mathcal{H}}(q+1)$ is less than $m_{\mathcal{H}}(N \leq q)$, the break point is $k = q + 1$ and the VC dimension is $d_{\text{VC}} = k - 1 = q$.

9. For hypothesis sets $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_K$ with finite, positive VC dimensions $d_{\text{VC}}(\mathcal{H}_k)$ (same input space $\mathcal{X}$), 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 $d_{\text{VC}}\left(\cap_{k=1}^{K} \mathcal{H}_k\right)$? (The VC dimension of an empty set or a singleton set is taken as zero.)

   Answer: [b] $\boldsymbol{0 \leq d_{\text{VC}}\left(\cap_{k=1}^{K} \mathcal{H}_k\right) \leq \min\{d_{\text{VC}}(\mathcal{H}_k)\}_{k=1}^{K}}$

   Since the intersection of the sets $\mathcal{H}_k$ can be an empty or singleton set, the lower bound must be zero.

   There is some flexibility in the upper bound. By definition, a hypothesis set or intersection set with a VC dimension of d_{VC} can shatter data sets with d_{VC} points.

   The loosest possible upper bound is $\sum_{k=1}^{K} d_{\text{VC}}(\mathcal{H}_k)$ since it will always be greater than the number of points the intersection set able to shatter.

   A tighter upper bound is $\max\{d_{\text{VC}}(\mathcal{H}_k)\}_{k=1}^{K}$ since it is the number of points that the most flexible hypothesis set can shatter. This would happen to be the tightest upper bound only when all hypothesis sets share the same d_{VC}, i.e., $d_{\text{VC}}(\mathcal{H}_1) = d_{\text{VC}}(\mathcal{H}_2) = \cdots = d_{\text{VC}}(\mathcal{H}_k)$.

   By the same logic, the tightest upper bound must be $\min\{d_{\text{VC}}(\mathcal{H}_k)\}_{k=1}^{K}$ because all hypothesis sets in the intersection set should, at a minimum, be able to shatter data sets with that many points.

10. For hypothesis sets $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_K$ with finite, positive VC dimensions $d_{\mathrm{VC}}(\mathcal{H}_k)$ (same input space $\mathcal{X}$), 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 $d_{\mathrm{VC}}\left(\cup_{k=1}^{K} \mathcal{H}_k\right)$?

    Answer: [e] $\max\{d_{\mathrm{VC}}(\mathcal{H}_k)\}_{k=1}^{K} \le d_{\mathrm{VC}}\left(\cup_{k=1}^{K} \mathcal{H}_k\right) \le K - 1 + \sum_{k=1}^{K} d_{\mathrm{VC}}(\mathcal{H}_k)$

    Following the train of thought used in problem 9, the lower bound of the union set is $\max\{d_{\mathrm{VC}}(\mathcal{H}_k)\}_{k=1}^{K}$ since it is always possible to shatter that many points by taking the most flexible hypothesis set, which has the highest VC dimension.

    One approach to find the upper bound is to determine the break point (and consequently, the VC dimension) using the growth function of the union set of two hypothesis sets and work recursively from that/use mathematical induction.

    For any two hypothesis sets $\mathcal{H}_i$ and $\mathcal{H}_j$ in the union set, the growth function is bounded by

$$m_{\mathcal{H}_i \cup \mathcal{H}_j}(N) \le m_{\mathcal{H}_i}(N) + m_{\mathcal{H}_j}(N) \le \sum_{k=0}^{d_{\mathrm{VC}}(\mathcal{H}_i)} \binom{N}{k} + \sum_{k=0}^{d_{\mathrm{VC}}(\mathcal{H}_j)} \binom{N}{k}$$

    since it cannot be more than the sum of the individual growth functions (limiting case where two subsets of a data set with VC dimensions of $d_{\mathrm{VC}}(\mathcal{H}_i)$ and $d_{\mathrm{VC}}(\mathcal{H}_j)$, respectively, are classified correctly exactly and only by the two hypothesis sets).

    Rewriting the second term using the equality $C_k^n = C_{n-k}^n$ and applying a change of variables $k' = N - k$, we get

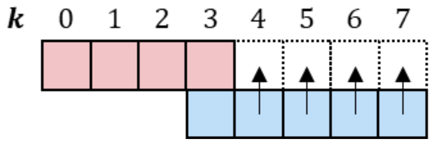
$$m_{\mathcal{H}_i \cup \mathcal{H}_j}(N) \le \sum_{k=0}^{d_{\mathrm{VC}}(\mathcal{H}_i)} \binom{N}{k} + \sum_{k=0}^{d_{\mathrm{VC}}(\mathcal{H}_j)} \binom{N}{N-k} = \sum_{k=0}^{d_{\mathrm{VC}}(\mathcal{H}_i)} \binom{N}{k} + \sum_{k'=N-d_{\mathrm{VC}}(\mathcal{H}_j)}^{N} \binom{N}{k'}$$

    Then, we can determine the break point by finding the N that gives $m_{\mathcal{H}_i \cup \mathcal{H}_j}(N) \le 2^N - f(N)$, where f is some arbitrary positive function.

The first summation becomes equal to 2^N when the C_k^N contributions from $k = d_{\mathrm{VC}}(\mathcal{H}_i) + 1$ to $k = N$ is added to it.

- When $N \leq d_{\mathrm{VC}}(\mathcal{H}_i) + d_{\mathrm{VC}}(\mathcal{H}_j)$, the second summation has more than enough terms to "transfer" to the first summation to change the upper bound of the latter to $k = N$. After this "move", the second summation still has remaining terms, making it a positive term (thus not satisfying the conditions for the $-f(N)$ term). An example with $N = d_{\mathrm{VC}}(\mathcal{H}_i) + d_{\mathrm{VC}}(\mathcal{H}_j)$ is shown in the schematic below, where each box is a combination term, and the top and bottom rows represent the first and second summations.

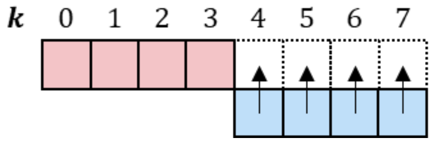
  $N = 7$, $d_{\mathrm{VC}}(\mathcal{H}_i) = 3$, $d_{\mathrm{VC}}(\mathcal{H}_j) = 4$

  If $d_{\mathrm{VC}}(\mathcal{H}_i)$ or $d_{\mathrm{VC}}(\mathcal{H}_j)$ increases, there will be more boxes on the second row left over and the second summation stays a positive term. Therefore, $N < d_{\mathrm{VC}}(\mathcal{H}_i) + d_{\mathrm{VC}}(\mathcal{H}_j)$ also holds.

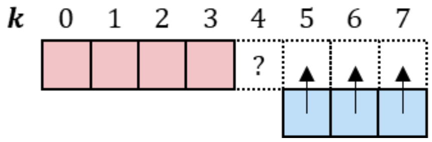
- When $N = d_{\mathrm{VC}}(\mathcal{H}_i) + d_{\mathrm{VC}}(\mathcal{H}_j) + 1$, the second summation has just enough terms that, when combined with the first summation, gives $\sum_{k=0}^{N} C_k^N = 2^N$.

  $N = 7$, $d_{\mathrm{VC}}(\mathcal{H}_i) = 3$, $d_{\mathrm{VC}}(\mathcal{H}_j) = 3$

- When $N \geq d_{\mathrm{VC}}(\mathcal{H}_i) + d_{\mathrm{VC}}(\mathcal{H}_j) + 2$, the second summation does not have enough terms to "complete" the first summation. An example is shown below for $N = d_{\mathrm{VC}}(\mathcal{H}_i) + d_{\mathrm{VC}}(\mathcal{H}_j) + 2$ only, but the same scenario arises when $N > d_{\mathrm{VC}}(\mathcal{H}_i) + d_{\mathrm{VC}}(\mathcal{H}_j) + 2$.

  $N = 7$, $d_{\mathrm{VC}}(\mathcal{H}_i) = 3$, $d_{\mathrm{VC}}(\mathcal{H}_j) = 2$

  As such, the missing terms come from outside the second summation and must be subtracted away in the end, i.e., $m_{\mathcal{H}_i \cup \mathcal{H}_j}(N) \leq \sum_{k=0}^{N} C_k^N - \sum_{k=\alpha}^{\beta} C_k^N = 2^N - f(N)$, where α and β are the smaller and larger, respectively, of $\{N - d_{\mathrm{VC}}(\mathcal{H}_j),\ N - d_{\mathrm{VC}}(\mathcal{H}_i) - 1\}$, and $f(N) = \sum_{k=\alpha}^{\beta} C_k^N$. Therefore, the break point is $k = d_{\mathrm{VC}}(\mathcal{H}_i) + d_{\mathrm{VC}}(\mathcal{H}_j) + 2$, and the corresponding VC dimension is $d_{\mathrm{VC}}(\mathcal{H}_i \cup \mathcal{H}_j) = k - 1 = d_{\mathrm{VC}}(\mathcal{H}_i) + d_{\mathrm{VC}}(\mathcal{H}_j) + 1$.

This means that the VC dimension of the union set of two hypothesis sets is bounded by

$$d_{\mathrm{VC}}\left(\mathcal{H}_i \cup \mathcal{H}_j\right) \leq d_{\mathrm{VC}}(\mathcal{H}_i) + d_{\mathrm{VC}}\left(\mathcal{H}_j\right) + 1$$

From the logic and schematics on the previous page, it is obvious that for each additional hypothesis set $\mathcal{H}_k$, the number of boxes (and consequently, the VC dimension) increases by $d_{\mathrm{VC}}(\mathcal{H}_k) + 1$.

For example, the upper bound on the VC dimension for a union set with three hypothesis sets is

$$d_{\mathrm{VC}}\left(\cup_{k=0}^{3} \mathcal{H}_k\right) \leq d_{\mathrm{VC}}(\mathcal{H}_1 \cup \mathcal{H}_2) + d_{\mathrm{VC}}(\mathcal{H}_3) + 1 = d_{\mathrm{VC}}(\mathcal{H}_1) + d_{\mathrm{VC}}(\mathcal{H}_2) + d_{\mathrm{VC}}(\mathcal{H}_3) + 2$$

By generalizing for a union set with K hypothesis sets, the upper bound on the VC dimension can be shown to be

$$d_{\mathrm{VC}}\left(\cup_{k=0}^{K} \mathcal{H}_k\right) \leq \sum_{k=0}^{K} d_{\mathrm{VC}}(\mathcal{H}_k) + K - 1$$