Pre-knowledge about VC Dimension — Lecture Notes of "Machine Learning Foundations" Course

Date: October 26, 2019

1 Training versus Testing

If $|\boldsymbol{H}| = M$ finite, N large enough, for whatever g picked by $\boldsymbol{A}$, $E_{out}(g) \approx E_{in}(g)$; if $\boldsymbol{A}$ finds one g with $E_{in}(g) \approx 0$, "Probably Approximately Correct" guarantees for $E_{out}(g) \approx 0$.

The following **Figure** 1 depicts the basic learning procedure:

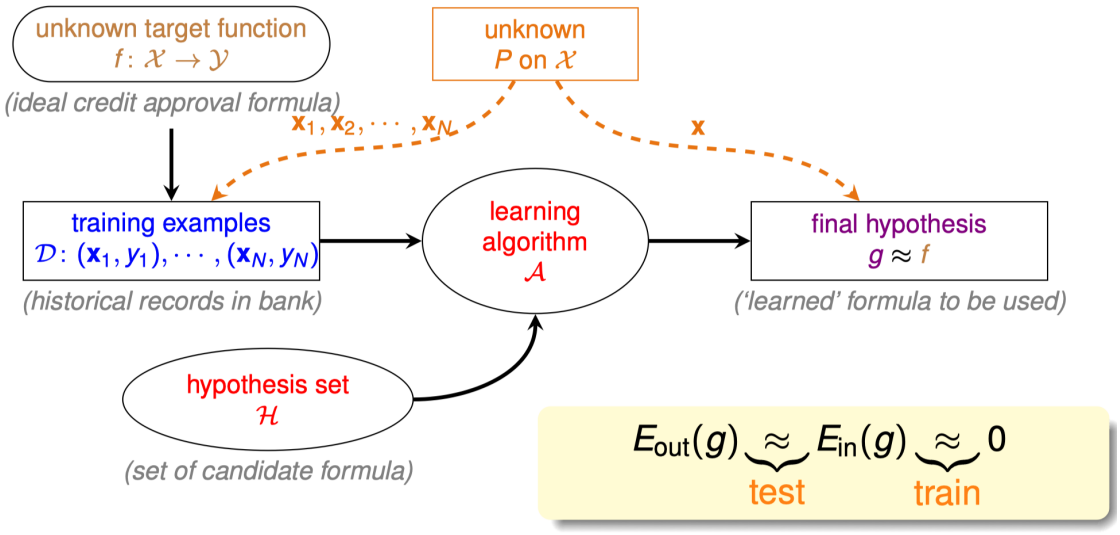


Figure 1: the flow chart of learning procedure

The algorithm $\boldsymbol{A}$ should be able to freely pick the equation that it thinks is most suitable in $\boldsymbol{H}$. Therefore, the most suitable equation may be any one of $\boldsymbol{H}$, possibly h_1, possiblyh_2, and any h of $\boldsymbol{H}$ may become g. But we know that our $\boldsymbol{D}$ is just a sample from the underlying population. Since it is a sample, there exists **sampling error**. For example, if we want to know the probability of throwing a coin ending up as a head, and we conduct 5 experiments in a test, there is a certain possibility of getting 5 heads in a row. At this time, the probability of getting head is 1, it may not be true. Therefore, this is a bad sample. *Any sample that differs greatly from the overall distribution due to sampling error can be called a* ***bad sample***.

Learning could also encounter the trouble of bad sample. For example, h_1 is actually a very good equation, which could have been g, but due to sampling error, it encountered a bad sample, which caused $E_{in}(h_1)$ to be large, and $\boldsymbol{A}$ did not choose it. Another example is that h_2 is a bad equation, and it encounters a bad sample. It happens that $E_{in}(h_2)$ is very small, causing $\boldsymbol{A}$ to choose it as g. Therefore, every h is likely to encounter this kind of trouble of bad sample. For any h, the bad sample will cause $|E\_\{in\}(h) - E\_\{out\}(h)| > \varepsilon$.

Therefore, as long as any h in $\boldsymbol{H}$ encounters a bad sample, $\boldsymbol{A}$ will encounter problems when selecting the model, and the learning may be bad. Then what is the probability of bad sample occurring:

$$
\begin{aligned}
&\mathbb{P}_D[BAD\ \mathcal{D}] \\
&= \mathbb{P}_D\left[BAD\ \mathcal{D} \text{ for } h_1 \text{ or } BAD\ \mathcal{D} \text{ for } h_2 \text{ or } \ldots \text{ or BAD } \mathcal{D} \text{ for } h_M\right] \\
&\leq \mathbb{P}_D\left[BAD\ \mathcal{D} \text{ for } h_1\right] + \mathbb{P}_D\left[BAD\ \mathcal{D} \text{ for } h_2\right] + \ldots + \mathbb{P}_D\left[BAD\ \mathcal{D} for h_M\right] \quad (1) \\
&\leq 2\exp\left(-2\epsilon^2 N\right) + \leq 2\exp\left(-2\epsilon^2 N\right) + \ldots + \leq 2\exp\left(-2\epsilon^2 N\right) \\
&= 2M\exp\left(-2\epsilon^2 N\right)
\end{aligned}
$$

It's obvious that whether the learning is good is related to the number of equations M in $\boldsymbol{H}$. When $M$is finite, the larger size of the sample data, the lower the probability of the occurance of a bad sample. Similarly, if M is too large, we are more likely to

encounter bad samples.

So if the number of equations in the hypothesis set $|H| = M$ is finite, and the sample size N is large enough, we'll have a high probability of not encounting the bad sample, that is, regardless of what g the algorithm $\boldsymbol{A}$ picks, $E_{out}(g) \approx E_{in}(g)$ will always be guaranteed. So learning can be focused on the following two issues:

- Ensure $E_{in}(g)$ and $E_{out}(g)$ are very close. (Whether the number of models in the Hypothesis Set is appropriate, and whether the amount of data used for training is large enough.)
- Ensure $E_{in}(g)$ is small enough. (Whether the number of models in the Hypothesis Set is too small.)

Here exists a dilemma:

1. If M is small, according to $\mathbb{P}_D[BAD\ D] \leq 2M \exp\left(-2\epsilon^2 N\right)$, the first requirement is "very close" and the hypothesis with small M does a good job. But for the second problem, it is difficult to find the smaller g of $E_{in}(g)$. (If the data is generated by a quadratic equation, only the linear equation can be selected in the Hypothesis Set.)
2. If M is large, we have more freedom to choose the model and end up with choosing the model with very low E_{in}. Yet according to the formula, the possibility of encountering the bad sample will be greatly increased.

Therefore, the further analysis will mainly focus on this M in the formula.

2 Effective Number of Hypotheses

The inequation (1) is derived under the assumption that the occurances of bad samples are independent, yet it's not usually the case.

Imagine two very similar equations $h_1 \approx h_2$. They encounter bad samples for events $B1$ and $B2$ respectively. Since these two models are very similar with each

other, $B2$ will also occur when $B1$ occurs, which is illustrated as **Figure** 2. It can be said that $B1$ and $B2$ have high degree of overlapping.

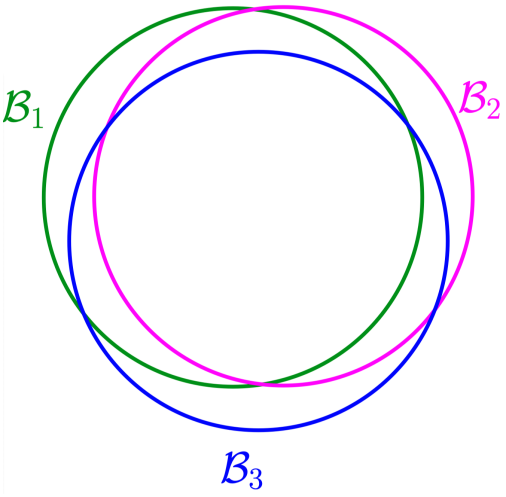


Figure 2: overlapping of similar hypotheses

We can treat the models that generate close results as a class.

For example, some models always have very similar predictions. As shown below in **Figure** 3, suppose our algorithm chooses a linear model as g on the plane, $\boldsymbol{H} = \{all\ lines\ in\ \mathrm{R}^2\}$. There are infinite numbers of models in H, yet we can classify these models into two categories. One is to classify x_1 as O, the other classifies x_1 as X.

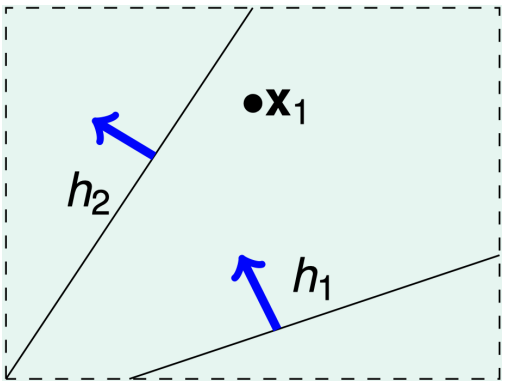


Figure 3: How many lines are there? (1)

If we have two data points x_1 and x_2, the straight lines in H can divide these two points x_1 and x_2 into 4 categories. Using these four types of lines to predict x_1 and

x_2, a total of four different results can be produced, which is illustrated in **Figure** 4.

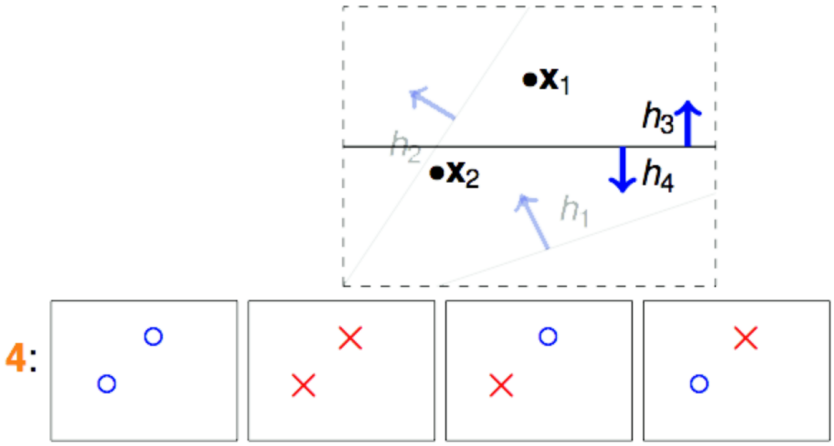


Figure 4: How many lines are there? (2)

If we have three data points x_1, x_2 and x_3, the first case is that these three points are not on a straight line, then the outcome is shown in **Figure** 5:

Figure 5: How many lines are there? (2-1)

Meanwhile, if the three points are in the same straight line, then the outcome is shown in **Figure** 6, which implies that the results are 'fewer than 8' when degenerate (e.g. collinearity or same inputs):

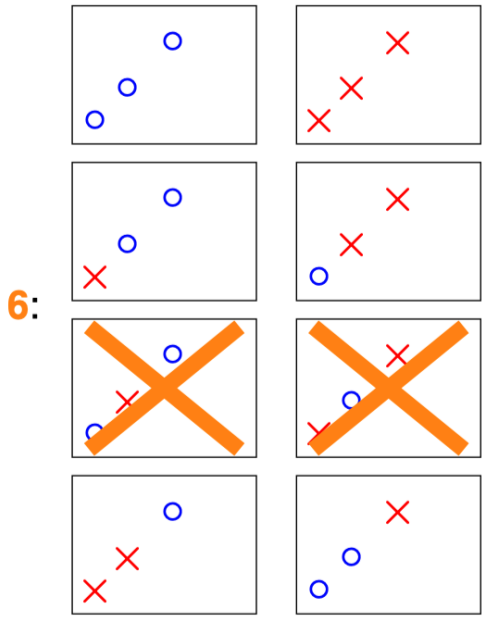


Figure 6: How many lines are there? (2-2)

Since most of the linear models produce the same (prediction) results, the different kinds of models are Class 2, Class 4 and Class 8 corresponding to the above examples (**Effective Number of Lines**). Similar lines will encounter or not encounter the bad sample at the same time. Since the previous union bound is based on the assumption of independence, the probability of $\boldsymbol{H}$ encountering the bad sample is obviously exaggerated. Therefore, we should rewrite the inequality as:

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

3 Dichotomies

Arbitrarily select an equation h from $\boldsymbol{H}$, let this h classify $\boldsymbol{D}$ into 2 categories, and output a result vector, such as predicting 4 points, output $\{o, o, o, x\}$ we call such an output vector a **dichotomy**.

It is not difficult to conclude that a linear equation corresponds to a dichotomy in $\boldsymbol{D}$, but a dichotomy corresponds to at least one linear equation. We consider all

the linear equations corresponding to a dichotomy as one class, and the effective number of lines is equal to the different numbers of dichotomies in different ***D***.

Obviously the number of dichotomy is smaller than or equal to the number of permutations and combinations of all data points. For example, the permutation combination corresponding to the picture with the orange cross in ***Figure*** 6 cannot be a dichotomy because they cannot be generated by any single linear model. (Of course, if you are not considering a straight line model, then that arrangement can be a dichotomy.)

So the effective(N)= how many different types of lines in the plane = how many different dichotomies can be produced when ***H*** is conducted in ***D***.

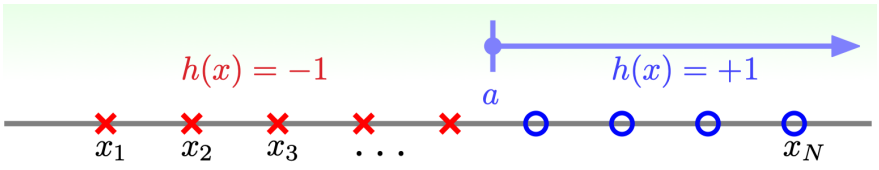
4 Growth Function

How many different types of dichotomy ***H*** could produce in ***D***? The number is related to ***H*** and also the size of the sample **N**. It can be expressed as:

$$\max |\mathcal{H}(x_1, x_2, \ldots, x_N)|$$

The above formula is also called a **growth function**. With a constant ***H***, the growth function is a function related to N. The following are the growth functions of several common Hypothesis Sets:

1. ***Positive Rays***



The input space is a one-dimensional real space. Predicted +1 if x is greater than a *threshold* a, otherwise predicted -1. For example: When N=4, Positive Rays acts on $x_1 \approx x_4$, which can produce 5 different dichotomies. As shown

below in **Figure** 7:

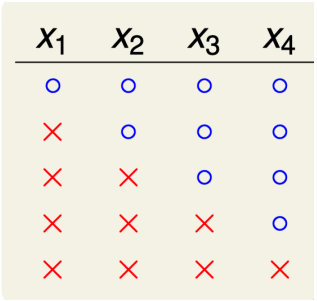
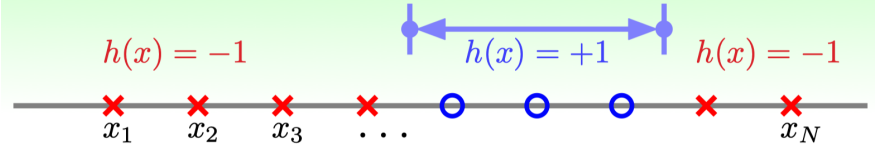


Figure 7: Dichotomies when N=4 under Positive Rays

It's growth function is $m_{\mathcal{H}(N)} = N + 1$.

2. ***Positive Intervals***



Similar with the Positive Rays, the input space is a one-dimensional real space, yet "Positive Intervals" has two thresholds, the prediction between the two thresholds is +1, and the remaining predictions are -1. For example: When N=4, Positive Intervals can produce 11 kinds of dichotomies for $x_1 \approx x_4$. As shown below in **Figure** 8:

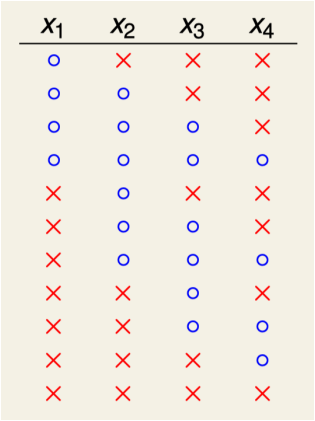
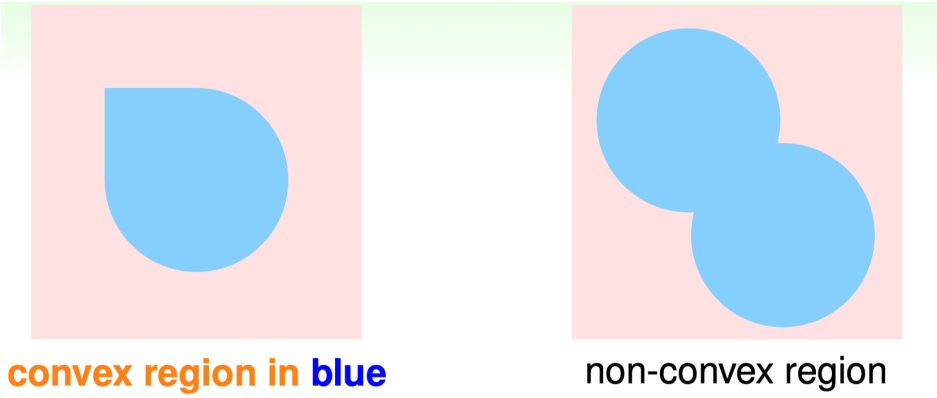


Figure 8: Dichotomies when N=4 under Positive Intervals

It's growth function is $m_{\mathcal{H}(N)} = \begin{pmatrix} N+1 \\ 2 \end{pmatrix} + 1 = \frac{1}{2}N^2 + \frac{1}{2}N + 1$.

3. ***Convex Sets***



The input space is a two-dimensional real space. ***H*** contains h which predicts +1 if and only if x is in a *convex region*, otherwise predicts -1.

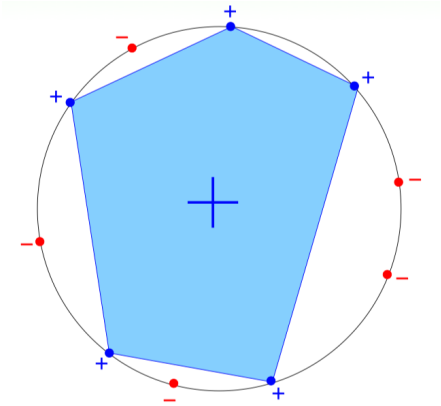


Figure 9: Dichotomies when N=4 under Convex Sets

Above **Figure** 9 depicts one possible set of N inputs: $x_1, x_2, \ldots, x_N$ on a big circle, and every dichotomy can be implemented by ***H*** using a convex region slightly extended from contour of positive inputs, under this circumstance, we call those **N inputs 'shattered' by *H***, i.e., It's growth function $m_{\mathcal{H}(N)} = 2^N \Leftrightarrow$ exists N inputs that can be shattered.

5 Break Point of H

Some $\boldsymbol{H}$'s growth functions are easy to find, such as the Positive Rays, Positive Intervals, and Convex Sets mentioned above; yet finding some other $\boldsymbol{H}$'s growth functions is not so easy, such as 2D perceptrons, we can't get its growth function directly, so we need to turn to the break point for $\boldsymbol{H}$:

If no k inputs can be shattered by $\boldsymbol{H}$, we call k a **break point for $\boldsymbol{H}$**. Some $\boldsymbol{H}$'s break point are listed below in Table 1.

Table 1: $m_{\mathcal{H}(N)}$ and Break Point for some $\boldsymbol{H}$

	$m_{\mathcal{H}(N)}$	Break Point
Positive Rays	$= N + 1 = O(N)$	2
Positive Intervals	$= \frac{1}{2}N^2 + \frac{1}{2}N + 1 = O(N^2)$	3
Convex Sets	$= 2^N$	0
2D Perceptrons	$< 2^N$ in some cases	4

Since we can't always know the growth function, the number of dichotomies we use to calculate the break point is still an estimation with bias. The biased estimator is actually the upper bound of dichotomies if we put H in D.

For example, with unknown $m_{\mathcal{H}(N)}$ when N = 3 and the break point k = 2, what's maximum possible dichotomies?

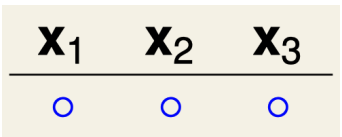


Figure 10: 1 Dichotomy

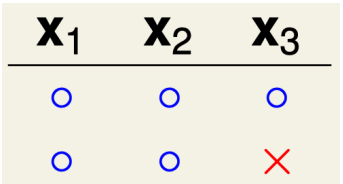


Figure 11: 2 Dichotomies

From above **Figure** 10 and **Figure** 11 we know that both 1 dichotomy and 2 dichotomies won't shatter any two points.

As below **Figure** 12 and **Figure** 13 show that 3 dichotomies won't shatter any two points, yet 4 dichotomies would.

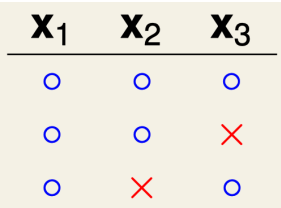


Figure 12: 3 Dichotomies

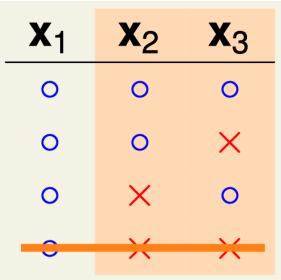


Figure 13: 4 Dichotomies (1)

As shown below **Figure** 14 and **Figure** 15, sometimes 4 dichotomies won't shatter two points, yet 5 dichotomies would.

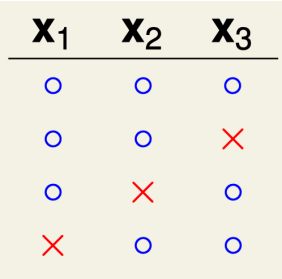


Figure 14: 4 Dichotomies (2)

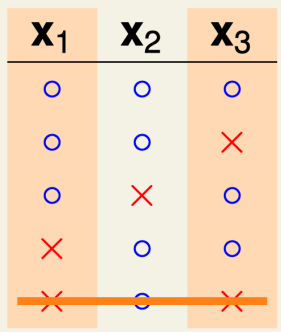


Figure 15: 5 Dichotomies (1)

As shown below **Figure** 16 and **Figure** 17, 5 dichotomies will always shatter 2 points.

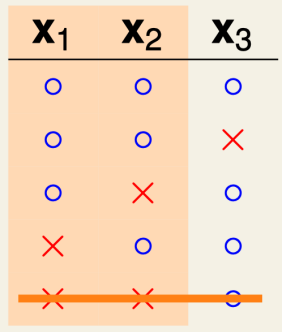


Figure 16: 5 Dichotomies (2)

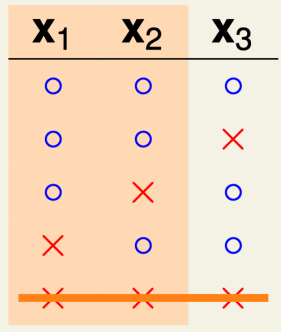


Figure 17: 5 Dichotomies (3)

So the upper bound of $N = 3, k = 2$ is 4. We use $B(N, k)$ to represent the upper limit of the number of dichotomies that can be generated by any $\boldsymbol{H}$ put in any D, with a break point of k and sample size N.

The conclusion we just obtained can be expressed as $B(3,2) = 4$. Since any 2 data points cannot be shatter, so when $N = 2$, $K = 2$, $B(2,2) < 4$, so $B(2,2) = 3$.

Although in many cases we can't get the growth function directly, yet if we know its break point, there is still a way to calculate the upper bound $B(N,k)$ of the growth function $m_{\mathcal{H}(N)}$. So we have a new goal, not to study the growth function directly, but to find **B(N,k)**.

It is known that $B(2,2) = 3$, $B(3,2) = 4$. So it's not difficult to know:

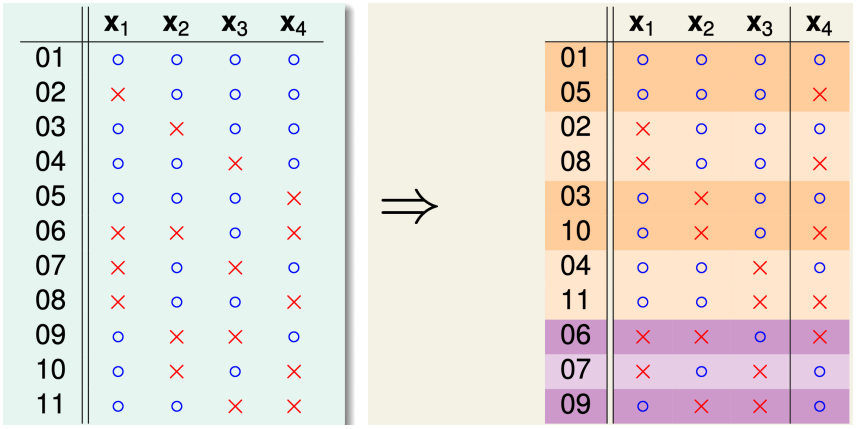
- When $k = 1$, one point (2 types of arrangement) has no way to shatter, so $B(?,1)$ is always equal to 1.
- When $k > N$, ***H*** can shatter N points, so the type of dichotomies it produces is equal to the number of all combinations of these N points 2^N.
- When $k = N$, remove one from the 2^N permutation combination, and the rest can be treated as dichotomies, so the number of dichotomies it produces can be $2^N - 1$ at most.

According to this, we get the **Figure** 18 below:

	$B(N,k)$	1	2	3	4	5	6	...
	1	1	2	2	2	2	2	...
	2	1	3	4	4	4	4	...
	3	1	4	7	8	8	8	...
N	4	1			15	16	16	...
	5	1				31	32	...
	6	1					63	...
	⋮	⋮						⋱

Figure 18: Table of Bounding Function (4/4)

After checking all 2^{2^4} sets of dichotomies, the $B(4,3)$ is 11 as below **Figure** 19 (orange: pair; purple: single).



	x_1	x_2	x_3	x_4
01	o	o	o	o
02	×	o	o	o
03	o	×	o	o
04	o	o	×	o
05	o	o	o	×
06	×	×	o	×
07	×	o	×	o
08	×	o	o	×
09	o	×	×	o
10	o	×	o	×
11	o	o	×	×

⇒

	x_1	x_2	x_3	x_4
01	o	o	o	o
05	o	o	o	×
02	×	o	o	o
08	×	o	o	×
03	o	×	o	o
10	o	×	o	×
04	o	o	×	o
11	o	o	×	×
06	×	×	o	×
07	×	o	×	o
09	o	×	×	o

Figure 19: Reorganized Dichotomies of B(4, 3)

As we see below **Figure** 20 and **Figure** 21, after the division, $B(4,3) = 11 = 2\alpha + \beta$.

	x_1	x_2	x_3
α	o	o	o
	×	o	o
	o	×	o
	o	o	×
β	×	×	o
	×	o	×
	o	×	×

Figure 20: Estimating Part of B(4, 3) (1)

	x_1	x_2	x_3	x_4
2α	o	o	o	o
	o	o	o	×
	×	o	o	o
	×	o	o	×
	o	×	o	o
	o	×	o	×
	o	o	×	o
	o	o	×	×
β	×	×	o	×
	×	o	×	o
	o	×	×	o

Figure 21: Estimating Part of B(4, 3) (2)

From $B(4,3)$ 'no shatter' any 3 inputs, we could have $\alpha + \beta$ 'no shatter' any 3, which means $\alpha + \beta \leq B(3,3)$. Similarly, we could also have α 'no shatter' any 2, which means $\alpha \leq B(3,2)$. To sum up, we have:

$$B(4,3) = 2\alpha + \beta$$

$$\alpha + \beta \leq B(3,3)$$

$$\alpha \leq B(3,2)$$

$$\Rightarrow B(4,3) \leq B(3,3) + B(3,2)$$

Therefore, we get the **Figure** 22 below, now we have the upper bound of the

bounding function:

	$B(N,k)$	k = 1	2	3	4	5	6
	1	1	2	2	2	2	2
	2	1	3	4	4	4	4
	3	1	4	7	8	8	8
N	4	1	≤ 5	11	15	16	16
	5	1	≤ 6	≤ 16	≤ 26	31	32
	6	1	≤ 7	≤ 22	≤ 42	≤ 57	63

Figure 22: Dichotomies of B(4, 3)

With mathematical induction, we can get:

$$B(N,k) \leq \sum_{i=0}^{k-1} \begin{pmatrix} N \\ i \end{pmatrix}$$

The upper bound of the growth function $B(N,k)$ is bounded, then the growth function itself is also bounded, so for the growth function with break point k, there exists $m_{\mathcal{H}} \leq \sum_{i=0}^{k-1} \begin{pmatrix} N \\ i \end{pmatrix}$, then we can apply these concepts in the VC Dimension.

References

[1] Hsuan-Tien Lin, "Machine Learning Foundations", Lecture 5 to 8.

[2] Beader's blog, http://beader.me/mlnotebook/

[3] Kubi Code's blog, http://kubicode.me/2015/08/15/Machine%20Learning/VC-Dimension/

[4] https://github.com/RedstoneWill/NTU-HsuanTienLin-MachineLearning