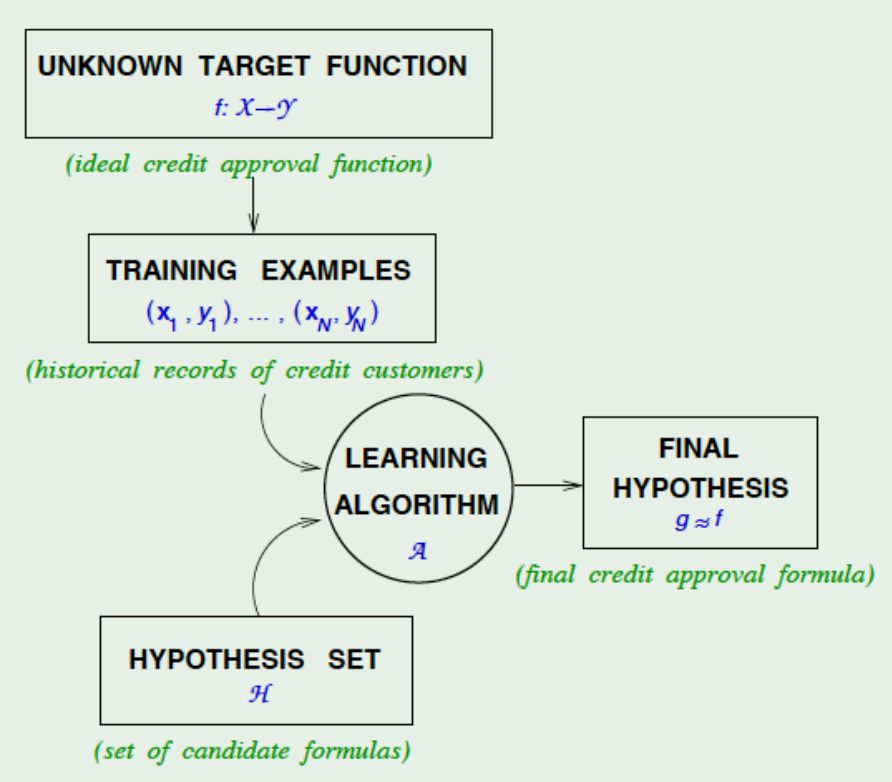
Theory of Generalization – A brief introduction

In ML, we are given a training set (say, of size N). We consider a **hypothesis set** H = {h1, h2, …}. The hypothesis set may be infinite as well, e.g., set of all lines in 2-d space.

We use a **learning algorithm** to select one hypothesis g from H, which we believe is the best one, i.e., its error on the training set $E_{in}(g)$ is the least. Note that selecting this 'best' hypothesis is analogous to learning the optimal values of the parameters. You can think that the learning algorithm takes as input (i) the training examples, and (ii) a hypothesis set, and gives as output the final hypothesis g having particular values of parameters (see the following diagram).



Once we have learned a hypothesis, we will apply it on the unknown test set. Let $E_{out}(g)$ be the error that we get on the test set.

Now, there are two fundamental questions in ML:

(1) How how can we learn a good model, i.e., make $E_{in}(g) \sim 0$? – this is the ALGORITHMS part.

(2) Can we guarantee that error on an unknown test set $E_{out}(g)$ will be close enough to $E_{in}(g)$? – this is the THEORY part. This theory is known as THEORY OF GENERALIZATION.

We need to formalize one error being "close enough" to the other. Formally, we would like to be able to say probability of $\{ \mid E_{in}(g) - E_{out}(g) \mid > \varepsilon \}$ <= a small, finite quantity. The **VC inequality** provides such a bound:

$$\mathbb{P}[\ |E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon\ ] \ \leq\ 4\ \ m_{\mathcal{H}}(2N)\ \ e^{-\frac{1}{8}\epsilon^2 N}$$

The Vapnik-Chervonenkis Inequality

Points to note:

- The bound on the right-hand side is a function of ε. The lesser we want ε to be, we will have to pay a higher price in terms of the training data size N. Also note that N appears on the right-hand side in the exponential part. This implies that, if we want to reduce ε, N has to increase exponentially.
- The other unfamiliar term on the right-hand side is a measure of the **complexity (or power) of the hypothesis set H**. Let us understand this term in detail.

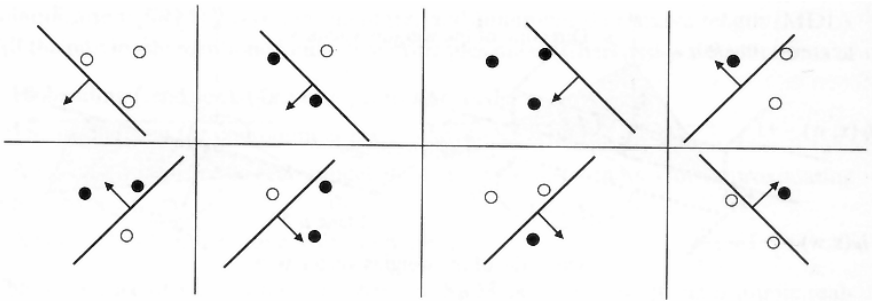
Let us assume we are given N points $x_1, x_2, \ldots, x_N$ in the training set, and the input space X is the 2-d plane (i.e., every point has only two features). For simplicity, let us consider a binary classification problem where classes are -1 and +1.

A hypothesis is a function h: X $\rightarrow$ {-1, +1}. There may be infinite hypotheses.
We define a **dichotomy** to be a mapping of the given N points to one of the classes. Mathematically, we have d: { $x_1, x_2, \ldots, x_N$ } $\rightarrow$ {-1, +1}. There can be at most 2^N dichotomies.

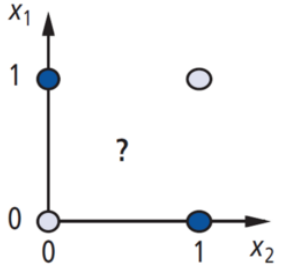
To get a measure of the complexity / power of a hypothesis set H, we ask the following question: For a given hypothesis set H and N points, how many of the 2^N dichotomies can H generate? More specifically, a configuration of N points is any placement of the N points on the plane. If a hypothesis set H is able to generate all dichotomies, then the following condition should hold - for *some* configuration of the N points, H should be able to, for *every* possible assignment of +1 and -1 to the points (in that configuration), perfectly partition the plane such that the positive points are separated from the negative points.

Note that the definition mentions 'some configuration', and not 'any configuration'. In other words, the N points can be placed as you like – you can configure the points such that you get the greatest number of dichotomies using the given hypothesis set.

Let us assume any linear model (perceptron, linear SVM, etc.) and N = 3 points. All 8 dichotomies can be achieved for N = 3 points. A bad choice will be to put all 3 points on a line, but we have the liberty of placing the points as we want.



However, for N=4 points, *not* all 16 dichotomies can be generated by a linear model. Only 14 dichotomies can be generated. If the points are assigned to -1 and +1 alternately, when considered in clockwise / anti-clockwise order, then we get a dichotomy that no linear model can generate (see figure below).



This observation agrees with the intuition that linear models are not the most powerful. A more powerful hypothesis set will be able to generate all 16 dichotomies for 4 points.

Going back to the VC inequality, the unfamiliar term on the right-hand side is actually a growth function to count the most dichotomies that H can generate over N points:

$$m_{\mathcal{H}}(N) = \max_{\mathbf{x}_1, \cdots, \mathbf{x}_N \in \mathcal{X}} |\mathcal{H}(\mathbf{x}_1, \cdots, \mathbf{x}_N)|$$

Some terms for characterizing power of a hypothesis set

If a hypothesis set H can generate all 2^N dichotomies for N points, we say that H **shatters** N points.

Breakpoint: if no data set of size k can be shattered by a hypothesis set H, then k is a breakpoint for H. E.g., for any linear model, k = 4.

VC dimension of a hypothesis set H is the largest value of N for which H can generate all 2^N dichotomies over N points (over some configuration of N points, not necessarily over any configuration of N points). The VC dimension is 1 less than a breakpoint. VC dimension of any linear model is 3.

Key result which allows generalization

This key result distinguishes between a hypothesis set having no breakpoint, and a hypothesis set having any break point:

$$\text{No break point} \implies m_{\mathcal{H}}(N) = 2^N$$

$$\text{Any break point} \implies m_{\mathcal{H}}(N) \text{ is } \mathbf{polynomial} \text{ in } N$$

Applying this result in the VC inequality: if we consider a hypothesis set having at least one break point, for a large enough N, the negative exponential term on the right-hand side will far exceed the polynomial term, and thus the right-hand side will ultimately be a small, finite quantity. Then we can argue that $E_{out}(g)$ will be sufficiently close to $E_{in}(g)$ with high probability.

Practical applications of VC dimension

If a model has more parameters, its VC dimension will usually be higher. Actually, VC dimension depends on the **effective** number of parameters of a hypothesis set. E.g., for SVM, we discussed that effective number of parameters is much lesser than actual number of parameters.

If we select a complex hypothesis set (having many parameters), then its VC dimension is likely to be higher, which implies that much larger training data would be needed to achieve the probability bound $P\{ | E_{in}(g) - E_{out}(g) | > \varepsilon$. Hence, the hypothesis set should be chosen keeping in mind the amount of training data available. If you have a relatively small amount of training data, then it is better to choose a simple hypothesis set having low number of parameters.

Note that VC dimension cannot be used to guarantee performance of a hypothesis set. We cannot claim things like "if one hypothesis set A has a lower VC dimension than another hypothesis set B, then A will perform better than B". VC dimension should only be used to argue about relative power of various hypothesis sets, to select which hypothesis set to use for a given training set size, etc.