Imperial College
London

EE3-23 Machine Learning

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

Assignment 2

Anand Kasture (CID: 00832896)

Date: February 20, 2017

Pledge

I, **Anand Kasture**, pledge that this assignment is completely my own work, and that I did not take, borrow or steal work from any other person, and that I did not allow any other person to use, have, borrow or steal portions of my work. I understand that if I violate this honesty pledge, I am subject to disciplinary action pursuant to the appropriate sections of Imperial College London.

1 Finite Realisable Hypothesis Set

Problem Definition

Consider a binary classification problem over some training sets $\mathcal{X}$ and $\mathcal{Y}$. Let $\mathcal{H}$ represent a finite hypothesis set such that $h : \mathcal{X} \to \mathcal{Y}$. The realisability assumption enforces the presence of some *perfect* hypothesis h^* that achieves zero out-of-sample error i.e. $R(h^*) = 0$. Under the independent and identically distributed (i.i.d) assumption on $\mathcal{X}$, the empirical error for a perfect hypothesis also equals zero i.e. $\hat{R}_n(h^*) = 0$.

Any Empirical Risk Minimisation (ERM) hypothesis within the hypothesis set is represented by g such that $g(x) \in \text{argmin } \hat{R}_n(g)$. The realisability assumption implies that $\hat{R}_n(g) = 0$ is true for $\forall g \in \mathcal{H}$. We are particularly interested in establishing a generic upper bound on the occurrence of the *bad* event $R(g) > \epsilon \ \forall g \in \mathcal{H}$. Namely, we will show that Equation 1 is satisfied with probability of at least $1 - \delta$.

$$R(g) \leq \frac{\log|\mathcal{H}|/\delta}{n} \tag{1}$$

Derivation

– Let $\mathcal{H}_B \subseteq \mathcal{H}$ represent the set of all bad hypotheses i.e. $\mathcal{H}_B = \{h \in \mathcal{H} : R(h) > \epsilon\}$. We can rewrite the probability of the bad event occurring as the probability of selecting a bad hypothesis.

$$\mathbb{P}[R(g) > \epsilon] = \mathbb{P}[g \in \mathcal{H}_B]$$

– The probability of selecting a bad ERM hypothesis g is upper bounded by the probability that there exists some hypothesis h with zero empirical error.

$$\mathbb{P}[g \in \mathcal{H}_B] \leq \mathbb{P}\Big[\exists h \in \mathcal{H}_B : \hat{R}_n(h) = 0\Big]$$

– The hypothesis h correctly classifies a single training sample with the probability $1-R(h)$. Therefore, since all the samples are i.i.d and $R(h) > \epsilon \ \forall h \in \mathcal{H}_B$, we may bound the probability of a selecting a fixed hypothesis with zero empirical error. Note: Using the $1 - \lambda \leq e^{-\lambda}$ result for the last inequality.

$$\mathbb{P}\Big[\hat{R}_n(h) = 0\Big] = (1 - R(h))^n \leq (1 - \epsilon)^n \leq e^{-\epsilon n}$$

– We will now invoke the *union bound* in order to extend the above result $\forall h \in \mathcal{H}_B$.

$$\mathbb{P}\Big[\exists h \in \mathcal{H}_B : \hat{R}_n(h) = 0\Big] \leq \sum_{h \in \mathcal{H}_B} \mathbb{P}\Big[\hat{R}_n(h) = 0\Big]$$

$$\sum_{h \in \mathcal{H}_B} \mathbb{P}\Big[\hat{R}_n(h) = 0\Big] \leq |\mathcal{H}_B| e^{-\epsilon n} \leq |\mathcal{H}| e^{-\epsilon n} = \delta$$

– Re-expressing ϵ in terms of the newly defined δ gives $\epsilon = \frac{\log|\mathcal{H}|/\delta}{n}$. Therefore, we complete the derivation as follows:

$$\mathbb{P}\left[R(g) > \frac{\log|\mathcal{H}|/\delta}{n}\right] \leq \delta \ \Rightarrow \ R(g) \leq \frac{\log|\mathcal{H}|/\delta}{n} \ \text{ with probability at least } (1 - \delta)$$

2 VC Dimension of Axis-Aligned Rectangles

In this section, we will show that the VapnikChervonenkis (VC) dimension of axis-aligned rectangles in the 2-D plane is precisely equal to 4. The hypothesis set $\mathcal{H}$ is defined for some $a_1 \leq a_2, b_1 \leq b_2$ as follows:

$$\mathcal{H} = \left\{ h : \mathbb{R}^2 \to \{0,1\} \ : \ \mathbb{I}(x \in R(a_1, a_2, b_1, b_2)) \right\}$$

The VC Dimension $d_{VC}$is defined as the largest number of points that a hypothesis set can shatter. The term shatter here refers to the generation of all possible classification permutations. For our binary classification problem through the use of axis-aligned rectangles, we will show that $d_{VC} \geq 4$ and $d_{VC} < 5$.

For $n = 1$ to $n = 4$ it is quite straightforward to conjure a 2-D arrangement that ensures the generation of all possible classification permutations. For instance, for $n = 4$, one may position the points such that they outline a parallelogram in order to demonstrate an arrangement that can be shattered with our hypothesis set. Note that while we can always also generate a 4 point arrangement (such as the set of co-linear points) that *cannot* be shattered, it is sufficient to find a single arrangement for some n that can be shattered in order to assert that $d_{VC} \geq 4$.

For $n = 5$, it can be seen that no possible arrangement can be shattered by considering the following argument: Assume that the rightmost, leftmost, highest and lowest points correspond to $+1$, whereas the last point corresponds to -1. Therefore, our rectangular classifier will completely encapsulate these 4 points as well as the remaining point that must lie somewhere in between the extremes. This is a general enough argument to prove that there will always be a 5 point classification permutation that cannot be achieved by a rectangle in 2-D space. Figure 1 illustrates an example to support this argument: we are unable to produce the $\{+1, +1, +1, +1, -1\}$ classification combination for any possible input arrangement when using axis-aligned rectangles. Here, the red circle is wrongly classified as -1.

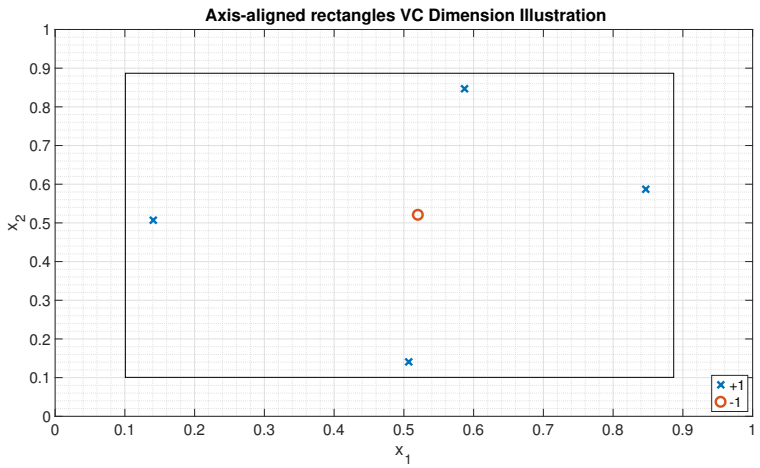

Figure 1: Axis-aligned rectangles VC Dimension Illustration

3 Polynomial Bounds on the Growth Function

The growth function $m_{\mathcal{H}}(n)$ is defined as the largest possible number of binary classification permutations on n points. This function plays an important role in achieving a finite RHS upper bound in the Vapnik-Chervonenkis inequality. Namely, the VC inequality implies that we can only acquire numerical guarantees on the difference between training and test performance when $m_{\mathcal{H}}(n)$ does not grow exponentially. It can be shown that the growth function is defined by polynomial expression when the VC dimension of the hypothesis set is finite. In this section, we will build upon Sauer's Lemma (Equation 2) in order to derive stricter polynomial upper bounds on $m_{\mathcal{H}}(n)$.

$$m_{\mathcal{H}}(n) \leq \sum_{j=0}^{d_{VC}(\mathcal{H})} \binom{n}{j} \tag{2}$$

Part A

Show $\forall n$, for some hypothesis class $\mathcal{H}$ with $d_{VC} < \infty$

$$m_{\mathcal{H}}(n) \leq n^{d_{VC}} + 1 \leq (n+1)^{d_{VC}}$$

– We will prove the above result through mathematical induction. The first step involves computing the base cases.

$$d_{VC} = 0 : \sum_{j=0}^{0} \binom{n}{j} \leq n^0 + 1 \longleftrightarrow 1 \leq 2$$

$$d_{VC} = 1 : \sum_{j=1}^{0} \binom{n}{j} \leq n^1 + 1 \longleftrightarrow n^1 + 1 \leq n^1 + 1$$

$$d_{VC} = 2 : \sum_{j=2}^{0} \binom{n}{j} \leq n^2 + 1 \longleftrightarrow \frac{n^2}{2} + \frac{n}{2} + 1 \leq n^2 + 1$$

– Assuming that the result is true for $d_{VC} \geq 2$

$$\sum_{j=0}^{d_{VC}} \binom{n}{j} = \sum_{j=0}^{d_{VC}-1} \binom{n}{j} + \binom{n}{d_{VC}} \leq n^{d_{VC}-1} + 1 + \binom{n}{d_{VC}}$$

– Using the result $\frac{n!}{(n-d_{VC})!} \leq n^{d_{VC}}$, we may simplify as follows:

$$n^{d_{VC}-1} + 1 + \binom{n}{d_{VC}} \leq n^{d_{VC}-1} + 1 + \frac{n^{d_{VC}}}{d_{VC}!}$$

– Because $\frac{1}{d_{VC}} < \frac{1}{2}$ for $d_{VC} > 2$

$$n^{d_{VC}-1} + 1 + \frac{n^{d_{VC}}}{d_{VC}!} \leq n^{d_{VC}-1} + 1 + \frac{n^{d_{VC}}}{2}$$

– Finally, we can use $\frac{1}{n} < \frac{1}{2} \longleftrightarrow \frac{n^{d_{VC}-1}}{n^{d_{VC}}} < \frac{1}{2} \longleftrightarrow n^{d_{VC}-1} < \frac{n^{d_{VC}}}{2}$ to show that

$$n^{d_{VC}-1} + 1 + \frac{n^{d_{VC}}}{2} \leq \frac{n^{d_{VC}}}{2} + 1 + \frac{n^{d_{VC}}}{2} \leq n^{d_{VC}} + 1$$

– Therefore, it follows that

$$m_{\mathcal{H}}(n) \leq \sum_{j=0}^{d_{VC}(\mathcal{H})} \binom{n}{j} \leq n^{d_{VC}} + 1 \leq (n+1)^{d_{VC}}$$

Part B

Show $\forall n \geq d_{VC}$, for some hypothesis class $\mathcal{H}$ with $d_{VC} < \infty$

$$m_{\mathcal{H}}(n) \leq \left(\frac{ne}{d_{VC}}\right)^{d_{VC}}$$

– Firstly, we will utilise Sauer's lemma in conjunction with the fact that $d_{VC} \leq n$ to bound the growth function.

$$\sum_{i=1}^{d_{VC}} \binom{n}{i} \leq \sum_{i=1}^{d_{VC}} \binom{n}{i} \left(\frac{n}{d_{VC}}\right)^{d_{VC}-i}$$

– Since $\frac{n}{d_{VC}} \geq 1$,

$$\begin{aligned}\sum_{i=1}^{d_{VC}} \binom{n}{i} \left(\frac{n}{d_{VC}}\right)^{d_{VC}-i} &\leq \left(\frac{n}{d_{VC}}\right)^{d_{VC}} \sum_{i=1}^{d_{VC}} \binom{n}{i} \left(\frac{d_{VC}}{n}\right)^{i} \\ &\leq \left(\frac{n}{d_{VC}}\right)^{d_{VC}} \left(1 + \frac{d_{VC}}{n}\right)^{n}\end{aligned}$$

– Using the very definition of the binomial theorem leads us

$$\left(1 + x\right)^{n} = \sum_{k=0}^{n} \binom{n}{k} x^{k}$$

– Finally, using the inequality $\left(1 + x\right)^{x} \leq e^{1}$, we obtain the final result

$$\begin{aligned}\left(\frac{n}{d_{VC}}\right)^{d} \left(1 + \frac{d_{VC}}{n}\right)^{n} &\leq \left(\frac{n}{d_{VC}}\right)^{d_{VC}} e^{d_{VC}} \\ &= \left(\frac{ne}{d_{VC}}\right)^{d_{VC}}\end{aligned}$$

4 Structural risk minimization

In this section, we will invoke the Perceptron Learning Algorithm (PLA) to classify points subject to a non-linear transformation. The overall goal is to study the effectiveness of the Structural Risk Minimisation (SRM) approach. This is an extension of the Empirical Risk Minimisation (ERM) method, except that we *choose* the best ERM hypothesis from an array of hypotheses sets by penalising overly complex sets.

Problem Definition

Let $\mathcal{X} = [0, 2.5] \times [-1, 2]$, and $f(x) = x(x-1)(x-2)$. We will be working with a 2-D noisy classification problem where x_1, x_2 are uniformly distributed on $\mathcal{X}$. The labels for any two points has the following distribution: $\mathbb{P}(y = 1 \mid x_2 \geq f(x_1)) = \mathbb{P}(y = -1 \mid x_2 < f(x_1)) = 0.9$. This set-up is illustrated in Figure 2.

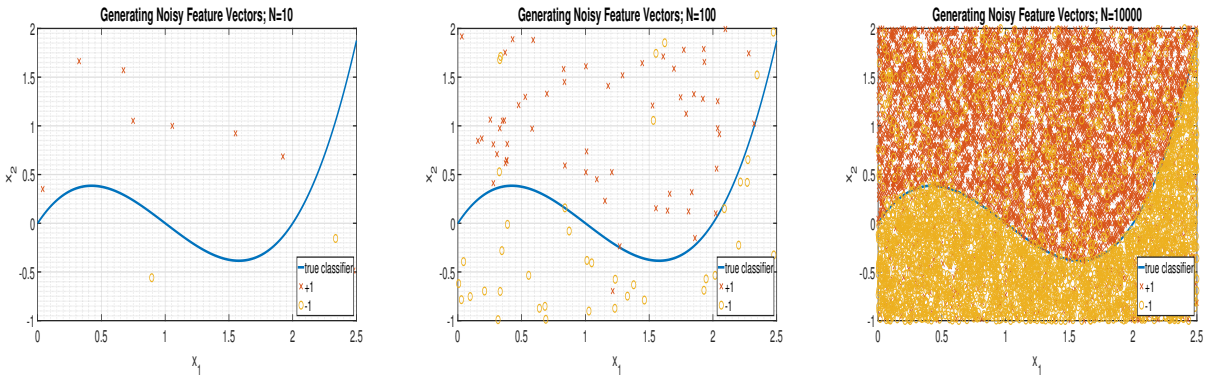

Figure 2: Generating Noisy Feature Vector Data for N=10,100,10000

Part B

In this sub-section, we will present training error and test error results for both the ERM and SRM approach. We will plot the *average* error trends for three training data sets of size $n \in [10, 100, 10000]$. In order to compute the SRM, we the hypothesis set $\mathcal{H}$ is denoted as a union of a specific set of classifiers: $\mathcal{H} = \bigcup_{q=0}^{4} \mathcal{H}_q$, where $\mathcal{H}_q = \{h : \mathbb{R} \to \mathbb{R}\}$ comprises of polynomials of degree at most q.

Note that the training data is randomly generated. This leads to a level of randomness across quantities such as the perceptron weights and the training error. Therefore, the following results have been acquired for 100 trials/repetitions. The test data set is generated using the same logic as that for the generation of training data, except that it comprises of a significantly larger number of input vectors in order to produce a reasonably accurate estimate of the theoretical error probability.

		Q				
		0	**1**	**2**	**3**	**4**
N	**10**	0.1210	0.0720	0.0490	0.0360	0.0340
	100	0.1912	0.1597	0.1330	0.1069	0.1114
	10000	0.1952	0.1871	0.1641	0.1085	0.1147

Table 1: ERM Average Training Errors

		Q				
		0	**1**	**2**	**3**	**4**
N	**10**	0.1896	0.2127	0.2294	0.2415	0.2696
	100	0.1299	0.1297	0.1048	0.0616	0.0686
	10000	0.1192	0.1118	0.0830	0.0110	0.0186

Table 2: ERM Average Test Errors

Table 1 and Table 2 show the average training errors and test errors that were recorded every hypothesis set $\mathcal{H}_q$ for the ERM method. We make the following observations: Training error generally increases with the number of samples. For a linearly non-separable data set, every perturbation of the linear classifier within the perceptron learning algorithm will lead to a large number of points that are newly misclassified as the grid is densely packed. Secondly, we see that the best training performance occurs for high polynomial orders. This is an expected result since the higher order polynomials are able to fit the training data much more closely.

Test error generally decreases with an increasing number of samples. This can be explained by Hoeffding's inequality, where the upper bound on the out-of-sample error is computed using a negative exponential that depends on the number of samples. The best test performance does not follow the same trends as those highlighted in Table 2. We can see that higher model orders perform better on larger sample sizes and vice versa. This is an expected result: simpler models often outperform overly complex ones for smaller sample sizes, and therefore one must choose different hypothesis classes in line with the size of the available sample size.

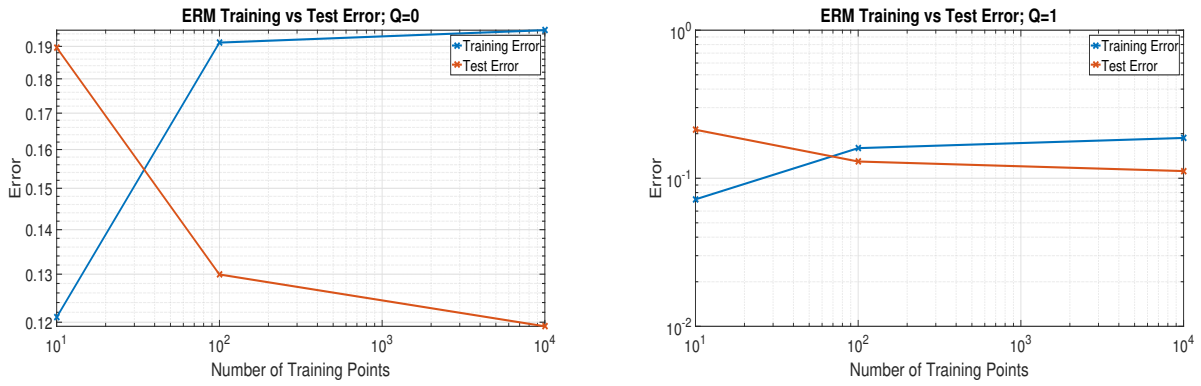

Figure 3: ERM Training vs Test Error for $Q = 0$ and $Q = 1$

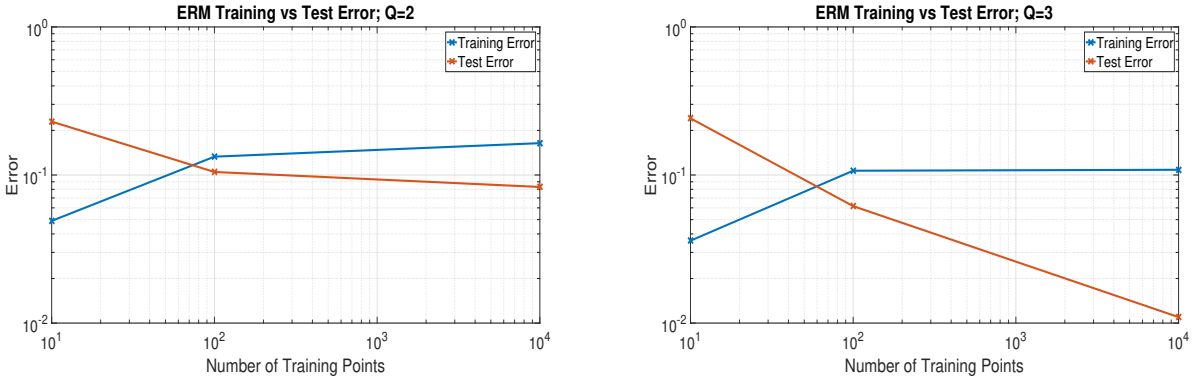

Figure 4: ERM Training vs Test Error for $Q = 2$ and $Q = 3$

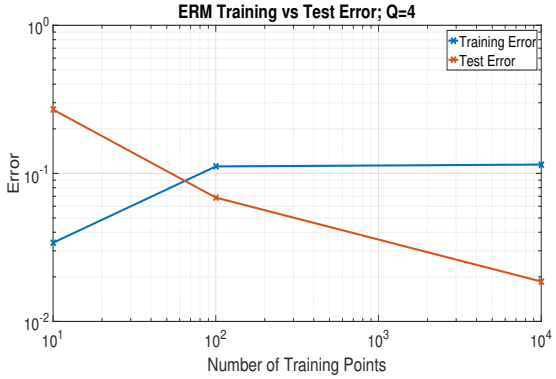

Figure 5: ERM Training vs Test Error for $Q = 4$

Figure 3, Figure 4 and Figure 5 illustrate the ERM training and test error (log-log) plots for every hypothesis set $\mathcal{H}_q$. These plots exhibit a clear trend: training error and the test error move in opposite directions as the number of training points increases.

Part C

In this sub-section, we will present results for the SRM method. Table 3 and Table 4 show the average training errors and test errors that were recorded for three different fractions of the complexity term. We make the following observations: While the SRM average training errors do outperform the best-case ERM values seen in Table 1, these are very minor improvements. We also note that the training error is indifferent to the heuristic constant. This is an expected result since the complexity term is only involved when computing the test error.

The SRM method, without any modifications to the complexity term, records comparatively poor test error values. On the contrary, we see much better test error values when we reduce the complexity term by a factor of 10 and 100. This is because of the way we compute the SRM test error in our MATLAB script: the complexity term is added on to the SRM training error (using the best ERM from one of the hypothesis sets). Furthermore, the complexity term is part of the VC inequality. This is an extremely pessimistic bound since it is derived using d_{VC}, which in turn is based on the set of points that are *easiest* to shatter.

		Heuristic Constant		
		1	**0.1**	**0.01**
N	**10**	0.0290	0.0290	0.0290
	100	0.1029	0.1029	0.1029
	10000	0.1085	0.1085	0.1085

Table 3: SRM Average Training Errors

		Heuristic Constant		
		1	**0.1**	**0.01**
N	**10**	3.3608	0.5262	0.2428
	100	1.4862	0.2061	0.0781
	10000	0.1989	0.0298	0.0129

Table 4: SRM Average Test Errors

Figure 6 and Figure 7 illustrate the SRM training and test error (log-log) plots for the three different heuristic constants. These plots exhibit the same trends in the test error that we have seen in the tables above.

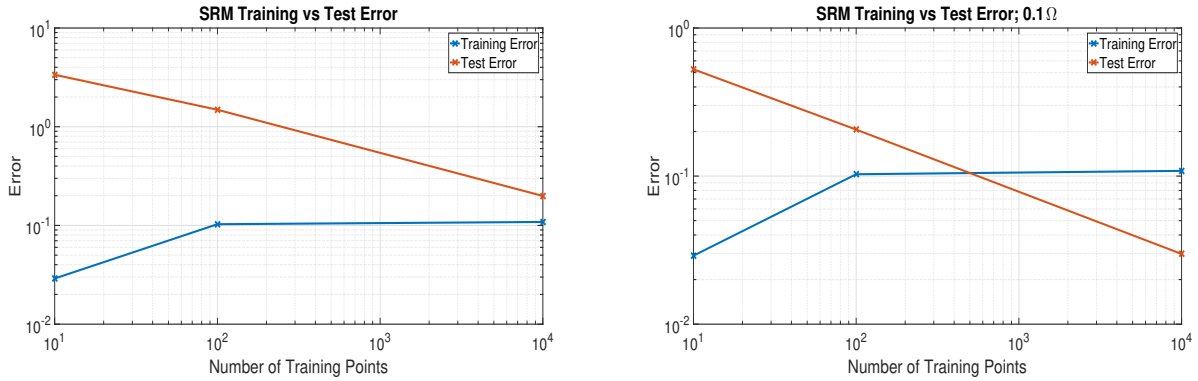

Figure 6: SRM Training vs Test Error for Heuristic Constant = 1 and 0.1

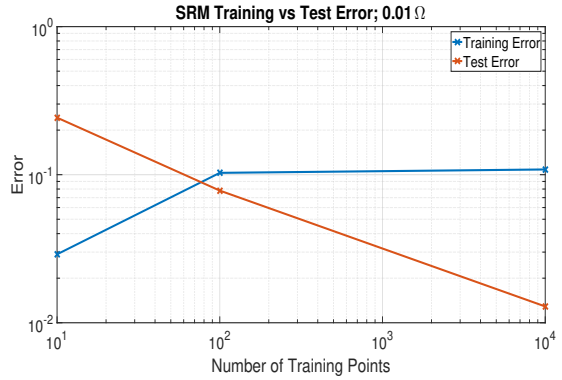

Figure 7: SRM Training vs Test Error for Heuristic Constant = 0.01