Mehryar Mohri
Foundations of Machine Learning
Courant Institute of Mathematical Sciences
Solution assignment 2
Solution of section B written by Cyril Allauzen.

A. VC Dimension

1. Let H and H' be two families of functions mapping from X to $\{0, 1\}$ with finite VC dimensions. Show that

$$\mathrm{VCdim}(H \cup H') \leq \mathrm{VCdim}(H) + \mathrm{VCdim}(H') + 1. \tag{1}$$

Use that to determine the VC dimension of the hypothesis set formed by the union of axis-aligned rectangles and triangles in dimension 2.

The number of ways m particular points can be classified using $H \cup H'$ is at most the number of classifications using H plus the number of classifications using H'. This gives immediately the following inequality for growth functions for any $m \geq 0$:

$$\Pi_{H' \cup H}(m) \leq \Pi_H(m) + \Pi_{H'}(m). \tag{2}$$

Let $\mathrm{VCdim}(H) = d$ and $\mathrm{VCdim}(H') = d'$. Then, by Sauer's lemma,

$$\Pi_{H' \cup H}(m) \leq \sum_{i=0}^{d} \binom{m}{i} + \sum_{i=0}^{d'} \binom{m}{i}. \tag{3}$$

Using the identity $\binom{m}{i} = \binom{m}{m-i}$ and a change of variable, this can be rewritten as

$$\Pi_{H' \cup H}(m) \leq \sum_{i=0}^{d} \binom{m}{i} + \sum_{i=0}^{d'} \binom{m}{m-i} \leq \sum_{i=0}^{d} \binom{m}{i} + \sum_{i=m-d'}^{m} \binom{m}{i}.$$

Now, if $m - d' > d + 1$, that is $m \geq d + d' + 2$,

$$\Pi_{H' \cup H}(m) \leq \sum_{i=0}^{m} \binom{m}{i} - \binom{m}{d+1} = 2^m - \binom{m}{d+1} < 2^m.$$

Thus, the VC dimension of $H \cup H'$ cannot be greater than or equal to $d + d' + 2$, which implies $\mathrm{VCdim}(H \cup H') \leq d + d' + 1$.

Now, by Lecture 3, the VC dimension of axis-aligned rectangles in dimension 2 is 4 and the VC dimension of triangles (3-gones) is 7. Thus, the VC dimension of the union of these sets is bounded by $4 + 7 + 1 = 12$.

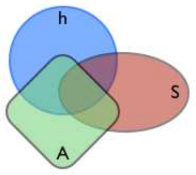

Figure 1: Illustration of $(h\Delta A)\cap S = (h\cap S)\Delta(A\cap S)$.

2. For two sets A and B, let $A\Delta B$ denote the symmetric difference of A and B, that is $A\Delta B = (A\cup B) - (A\cap B)$. Let H be a non-empty family of subsets of X with finite VC dimension. Let A be an element of H and define $H\Delta A = \{X\Delta A \colon X \in H\}$. Show that

$$\text{VCdim}(H\Delta A) = \text{VCdim}(H). \tag{4}$$

Fix a set S. We can show that the number of classifications of S using H is the same as when using $H\Delta A$. The set of classifications obtained using H can be identified with $\{S\cap h \colon h\in H\}$ and the set of classifications using $H\Delta A$ can be identified with $\{S\cap(h\Delta A)\colon h\in H\}$. Observe that for any $h\in H$,

$$S\cap(h\Delta A) = (S\cap h)\Delta(S\cap A). \tag{5}$$

Figure 1 helps illustrate this equality in a special case. Now, in view of this inequality, if $S\cap(h\Delta A) = S\cap(h'\Delta A)$ for $h, h'\in H$, then

$$(S\cap h)\Delta B = (S\cap h')\Delta B, \tag{6}$$

with $B = S\cap A$. Since two sets that have the same symmetric differences with respect to a set B must be equal, this implies

$$S\cap h = S\cap h'. \tag{7}$$

This shows that ϕ defined by

$$\begin{aligned}\phi\colon S\cap H &\to S\cap(H\Delta A)\\ S\cap h &\mapsto S\cap(h\Delta A)\end{aligned}$$

is a bijection and thus that the sets $S\cap H$ and $S\cap(H\Delta A)$ have the same cardinality.

B. SVMs

1. Download and install the `libsvm` software library from:
`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

2. Download the `ISOLET` data set from `http://archive.ics.uci.edu/ml/datasets/ISOLET`. There are two files: `isolet1+2+3+4`, and `isolet5`. `isolet1+2+3+4` should be used for training and validation, `isolet5` for testing only.

   The dataset corresponds to a number of people pronouncing each of the 26 letters in the alphabet. Each person pronounces the whole alphabet twice. We will consider the binary classification that consists of distinguishing the first 13 letters from the last 13 letters. Thus, assign label '-1' to the first 13, '+1' to the rest.

3. Normalize all input vectors (`isolet1+2+3+4` and `isolet5`): compute the scaling and offset on the `isolet1+2+3+4` so that each feature has zero mean and standard deviation 1, and apply the same scaling on the `isolet5` data.

   Using `svm-scale` would not have been the right thing here.

4. Split the data `isolet1+2+3+4`, containing the alphabet spoken 240 times, into 10 folds, ensuring that the same speaker is only in one of the folds by splitting after each 24 pronunciations (modulo a few that are missing; check that all your files start with the label '1').

   The key here is to ensure that each speaker is assigned to the same fold and to be careful with the missing datapoints. Using the `-v` option of the `svm-train` command-line utility would not do the right thing. We proceeded in two steps:

   (a) identify the speaker corresponding to each line of the data file and

   (b) randomly assign each speaker to one the folds and assign each line to random fold corresponding to the speaker for that line.

5. Let $x_1, \ldots, x_m$ denote the sample formed by the ten folds. For each feature f, let $f_1, \ldots, f_m$ denote its values for $x_1, \ldots, x_m$, and let $y_1, \ldots, y_m$ denote the labels. Compute the empirical correlation of each non-constant feature f with the labels

$$\widehat{\rho}(f, y) = \frac{\widehat{\sigma}_{fy}}{\sqrt{\widehat{\sigma}_{ff}\widehat{\sigma}_{yy}}},$$

   where $\widehat{\sigma}_{fy} = \frac{1}{m}\sum_{i=1}^{m}(f_i - \overline{f})(y_i - \overline{y})$, with $\overline{f}$ the average value of f, and $\overline{y}$ the average value of y. $\widehat{\sigma}_{ff}$ and $\widehat{\sigma}_{yy}$ are defined in a similar way. Sort all features in decreasing order of the absolute value of the correlation and save the correlation values.

   The 10 features which correlations are the largest in absolute value are given below.

Feature	Correlation
480	-0.306521
214	-0.287193
182	-0.280878
215	-0.280177
183	-0.275566
213	-0.272187
181	-0.271201
184	-0.265968
216	-0.261865
188	-0.260272

6. We first consider a *Naive* algorithm. Given a kernel K, this algorithm assigns a label to a new point simply based on its similarity with respect to the set of positively labeled versus its similarity with the negatively labeled points of the training set. Thus, for any point x, if we denote by $\mathbf{y}$ the vector of the labels in the training set and use the notation $\mathbf{K}_x = \begin{bmatrix} K(x,x_1) \\ \vdots \\ K(x,x_m) \end{bmatrix}$, the label it assigns to x is

$$h(x) = \operatorname{sgn}(\mathbf{K}_x \cdot \mathbf{y}).$$

Determine and report the performance of the Naive algorithm on the test set when using a polynomial kernel of degree d with $d = 1, 2, 3, 4$, when using a percentage p of the most correlated features, with $p = 100\%, 80\%, 40\%, 20\%$.

Polynomial kernels are of the form

$$(\gamma \cdot \mathbf{x}^T \mathbf{y} + c_0)^d. \tag{8}$$

By default when using the `svm-train` command-line utility, the value of γ is $1/F$ where F denotes the number of features and the value of c_0 is 0. We chose to use these default values of γ and c_0. The performance of the naive method on the test set is given below.

		Percentage of features			
Accuracy		20%	40%	80%	100%
Degree	1	66.39%	66.65%	67.60%	67.35%
	2	51.19%	55.23%	60.36%	64.34%
	3	68.31%	69.85%	72.80%	73.57%
	4	52.79%	56.76%	65.55%	70.43%

With $\gamma = 1/F$ and $c_0 = 0$, the performance of the even degree kernels is significantly worse than the one of the odd degree kernels. However, using $c_0 = 1$ would have made the performance of the even degree kernels match the one of the odd degree kernels.

7. Determine and report the performance of SVMs for the same set of kernels and the same sets of features. To determine the trade-off parameter C, use 10-fold cross validation with the ten folds previously defined (let the other parameters of polynomial kernels in `libsvm`, γ and c, be equal to their default values 1). Give a plot comparing the performance of SVMs with that of the Naive algorithm for each value of p.

   Depending on the choice of γ and c_0, the value of the best C can be quite different. We report here results obtained using $\gamma = 1/F$ and $c_0 = 0$.

% of features	Degree	C	Accuracy	
			(cross-validation)	(test set)
20%	1	0.4	78.17%	77.49%
	2	7.5	86.08%	85.57%
	3	7.5	87.72%	86.20%
	4	16	86.32%	84.99%
40%	1	3	81.27%	81.65%
	2	3	91.59%	90.70%
	3	8	92.59%	92.11%
	4	16	91.62%	89.80%
80%	1	10	85.50%	85.63%
	2	15	94.99%	94.36%
	3	4	95.71%	95.83%
	4	16	94.62%	93.77%
100%	1	4	85.94%	87.30%
	2	4	95.80%	94.87%
	3	6	96.09%	96.66%
	4	20	95.49%	94.55%

   Figure 2 gives a plot comparing the performance of SVMs with that of the Naive algorithm for each value of p.

8. Now, first multiply each feature f by its empirical correlation $\widehat{\rho}(f, y)$ and retrain SVMs with the full set of features. Report the test results obtained for the four values of d.

   The results obtained here seem to depend heavily on the polynomial kernel parameters γ and c_0. Ideally, one would like to determine the value of C using cross validation as done above. The results obtained using $C = 1$ are given below for three sets of polynomial kernel parameters.

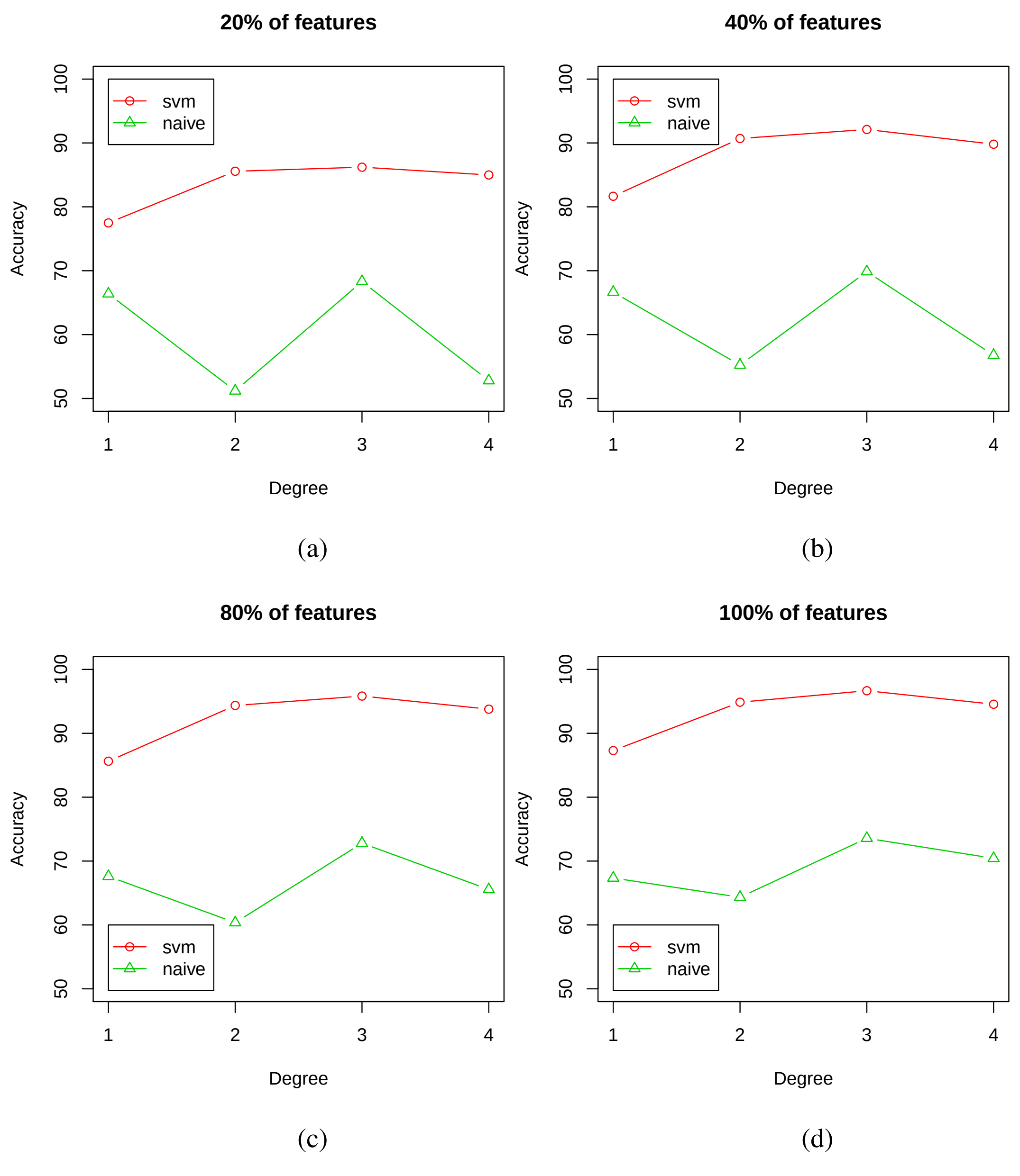

Figure 2: Accuracy on the test set of SVMs and the naive algorithms when using the (a) 20%, (c) 40%, (d) 80% and (d) 100% most correlated features

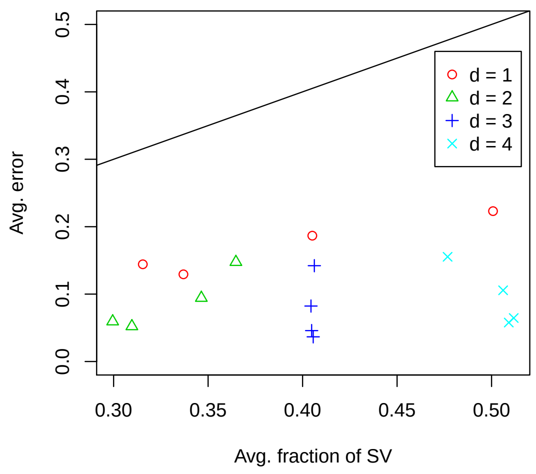

Figure 3: Average test error as a function of the fraction of support vector

$\gamma = 1/F, c_0 = 0$		$\gamma = 1/F, c_0 = 1$		$\gamma = 1, c_0 = 1$	
Degree	Accuracy	Degree	Accuracy	Degree	Accuracy
1	74.66%	1	74.66%	1	86.15%
2	50.03%	2	77.68%	2	93.33%
3	50.03%	3	79.03%	3	94.87%
4	50.03%	4	80.05%	4	94.55%

9. For each d and p, fix C to its best value obtained in question 7. For this value of C, for each d and p, train ten models, each time by excluding one of the ten folds. Compute the average number of support vectors and the average test error for each d and p. Plot the average error as a function of the average fraction of support vectors. Discuss your results and compare with the leave-one-out theorem presented in class.

   The average test error and the average number of support vectors are given below for the different values of p and d.

% of features	Degree	Avg. accuracy	Avg. error	Avg. number of SV
20%	1	77.69%	22.31%	3123.8
	2	85.25%	14.75%	2275.1
	3	85.79%	14.21%	2534.0
	4	84.46%	15.54%	2973.9
40%	1	81.33%	18.67%	2527.3
	2	90.57%	9.43%	2161.0
	3	91.77%	8.23%	2522.9
	4	89.44%	10.56%	3157.1
80%	1	85.57%	14.43%	1967.8
	2	94.06%	5.84%	1868.3
	3	95.41%	4.59%	2525.1
	4	93.55%	6.45%	3192.2
100%	1	87.06%	12.94%	2101.9
	2	94.78%	5.22%	1931.7
	3	96.34%	3.66%	2530.3
	4	94.22%	5.78%	3175.8

Figure 3 shows the average test error as a function of the fraction of support vectors. Observe that the average test error appears to be bounded by the average fraction of support vectors, in line with what was suggested by the leave-one-out analysis presented in class.