CS 6350: Machine Learning Fall 2016

Gopal Menon
Homework 5

November 15, 2016

1 Warm up: Margins

1. [5 points] Suppose we want to use an SVM to learn the XOR function in two dimensions. We know that XOR is not linearly separable, so we apply a feature transformation. In order to do so, we map the input $[x_1, x_2]$ into a space consisting of two features: x_1 and x_1x_2. All examples are Boolean (1 for positive and -1 for negative) What is the maximal margin? Draw the separating line back in original Euclidean input space.

x_1	x_2	x_1x_2	*Label*
-1	-1	$+1$	-1
-1	$+1$	-1	$+1$
$+1$	-1	-1	$+1$
$+1$	$+1$	$+1$	-1

XOR Function truth table

The XOR function truth table shows the original inputs x_1 and x_2 along with the new feature $x1x2$. The label column shows $x_1 \oplus x_2$. Figure 1 shows the new space with features x_1 and x_1x_2. The points in this input space are shown along with their labels. The axis for the feature x_1 is the linear separator with the maximum margin and is shown as the bold line. The four features in the new space become support vectors and the margin is half the distance between the lines joining the support vectors.

The input features x_1 and x_2 in the original feature space are shown in figure 2. The lines joining the support vectors now intersect each other. The line of maximum margin in figure 1 was the one where x_1 varied from $-\infty$ to $+\infty$ and x_1x_2 was 0. The corresponding line in the original feature space would be the one where x_1 varies from $-\infty$ to $+\infty$ and x_2 is 0, as $x_2 = 0$ corresponds to the case where $x_1x_2 = 0$ in the new feature space. This line of maximum separation is shown in bold in figure 2.

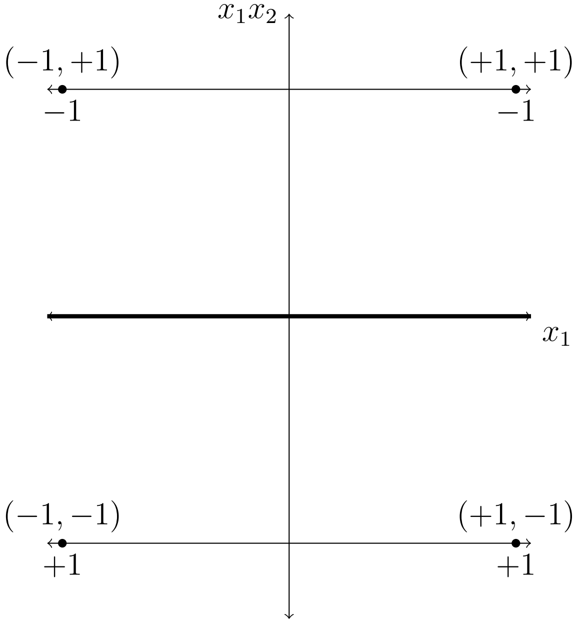

Figure 1: Separation of XOR function in feature space x_1 and x_1x_2.

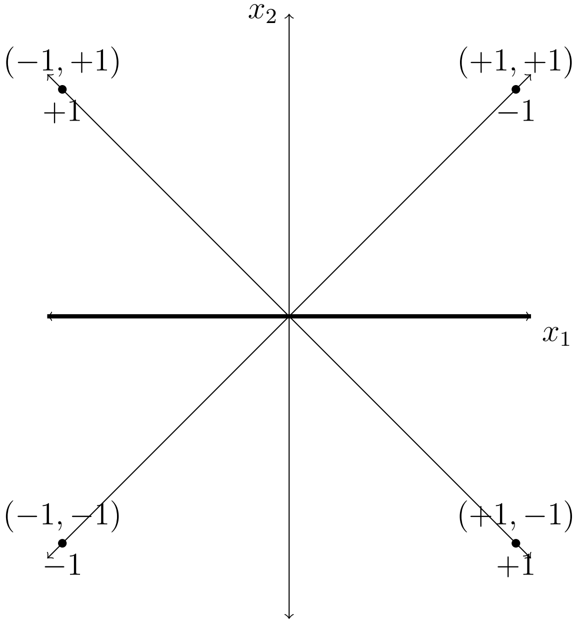

Figure 2: Separation of XOR function in original feature space x_1 and x_2.

2. [10 points] Consider the following collection of points:

Point	coordinate	label	Point	coordinate	label
x_1	(0, 0)	+	x_5	(1, 0)	-
x_2	(0, 1)	+	x_6	$(\frac{1}{2},\ \frac{\sqrt{3}}{2})$	-
x_3	(1, 1)	+	x_7	$(\frac{3}{2},\ 0)$	-
x_4	$(\frac{1}{2},\ 0)$	+	x_8	$(1,\ \frac{1}{2})$	-

Table 1: A collection of points

Suppose we have three training sets comprising of subsets of these points. We have

$$D_1 = \{x_1, x_2, x_3, x_5, x_7\}$$

$$D_2 = \{x_1, x_5, x_6, x_8\}$$

$$D_3 = \{x_3, x_4, x_5, x_7\}$$

(a) [6 points] Give the maximum possible margin for D_1, D_2 and D_3.

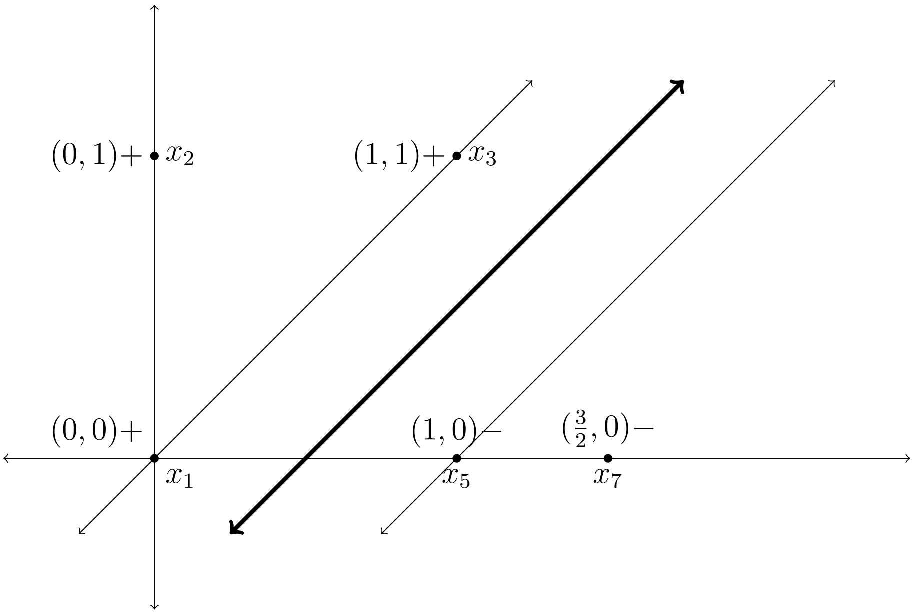

Figure 3: Points in set D_1.

The line separating the positive and negative examples with the maximum margin for D_1 is given is shown by the bold line in figure 3. This bold line is in the middle of the widest strip separating the positive and negative labels. This widest strip is shown by the two lines through x_1, x_3 and through x_5. The bold line intersects the horizontal axis at $(\frac{1}{2}, 0)$. All the separating lines are at 45° to the horizontal axis since one separating line passes through $(0,0)$ and $(1,1)$ and the three separating lines are parallel. So the distance between the bold line and the line through x_1 is $\frac{1}{2}\cos(45°) = \frac{1}{2}\frac{1}{\sqrt{2}} = \frac{1}{2\sqrt{2}} = 0.3536$

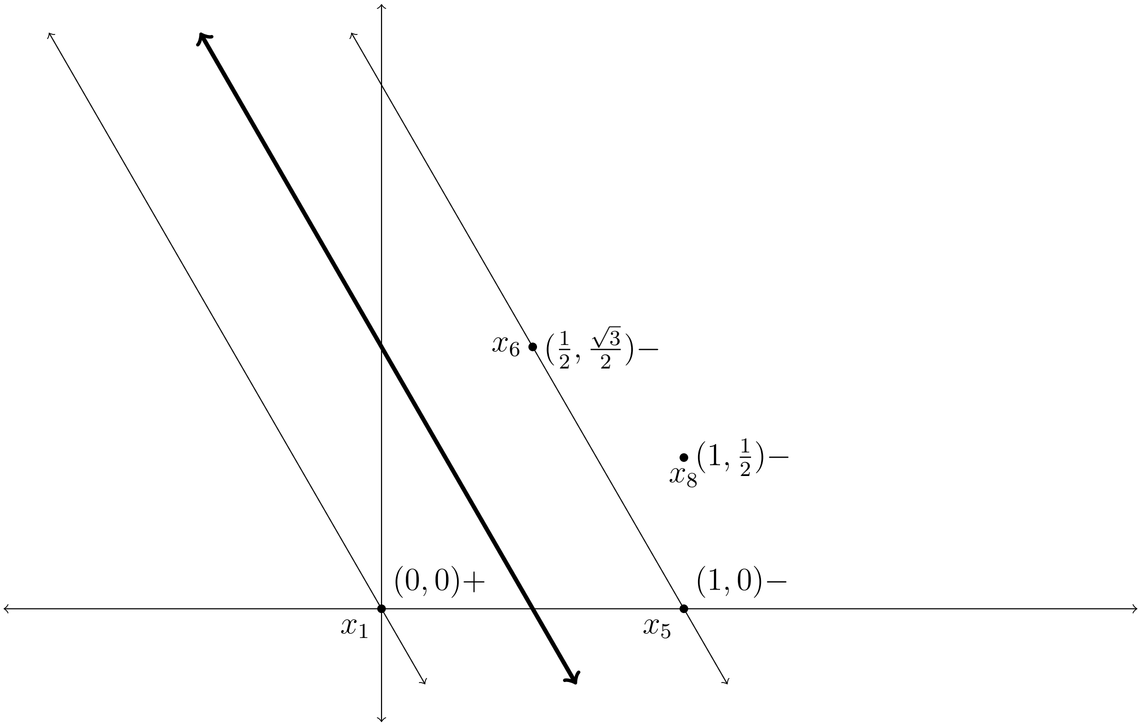

Figure 4: Points in set D_2.

The widest strip separating the two labels in D_2 is shown in figure 4, with the solid line marking the separating margin between the two labels. One side of the widest strip passes through x_5 and x_6 and the other is parallel and passes through $(0,0)$. The slope of the separating lines is $\frac{\frac{\sqrt{3}}{2}-0}{\frac{1}{2}-1} = -\sqrt{3}$ (the slope of the line joining x_5 and x_6). Based on the value of the slope, we can say that the lines make an angle of 120° with the horizontal. Which means that $\sin(180-120) = \sin(60) = \frac{\sqrt{3}}{2} = \frac{margin}{0.5}$ since the distance between the horizontal intercepts of the bold line and the line through x_6 and x_5 is 0.5. This means that $margin = \frac{\sqrt{3}}{4} = 0.4330$.

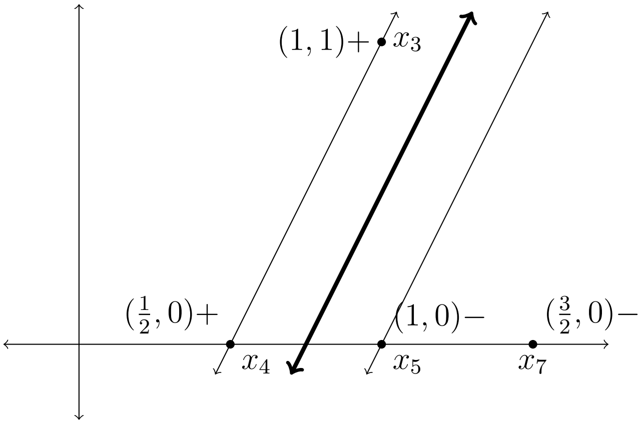

Figure 5: Points in set D_3.

The widest strip separating the two labels in D_3 is shown in figure 5, with the solid line marking the separating margin between the two labels. The separating lines have a slope of $\frac{1-0}{1-\frac{1}{2}} = \frac{1}{\frac{1}{2}} = 2$. The angle that the separating lines make with the horizontal is $\arctan(2)$. Since the horizontal component of the margin is $\frac{1}{4}$, the margin is $\frac{1}{4}\sin(\arctan(2)) = \frac{1}{4}\sin(1.1071) = \frac{1}{4} \times 0.8944 = 0.2236$.

(b) [2 points] What is the Perceptron mistake bound for these dataset. Which has the greatest Perceptron mistake bound.

Data set	Farthest point	R	Margin γ	Mistake Bound $\frac{R^2}{\gamma^2}$
D_1	x_7	1.5000	0.3536	17.9952
D_2	x_8	1.1180	0.4330	6.6667
D_3	x_7	1.5000	0.2236	45.0027

Table 2: Perceptron mistake bound for data sets D_1, D_2 and D_3

Data set D_3 has the greatest Perceptron mistake bound.

(c) [2 points] Rank the datasets in terms of "ease of learning". Justify your answer.

The mistake bound for a perceptron is the most number of mistakes a learning algorithm will make in finding a separator. So data sets with the minimum mistake bound should be easiest to learn as they have a lower limit on the number of mistakes that can be made. Therefore the data sets ranked by the ease of learning, starting with the easiest first are $\{D_2, D_1, D_3\}$.

2 Kernels

1. [15 points] If $K_1(\mathbf{x}, \mathbf{z})$ and $K_2(\mathbf{x}, \mathbf{z})$ are both valid kernel functions. In this question, you will prove that certain functions of kernels are valid kernels.

[Hint: For both the proofs below, use the the definition of a kernel as a dot product in a high dimensional space.]

(a) [5 points] Show that the product of two kernels is a kernel. That is, show that K in the expression below is a kernel:

$$K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z}) K_2(\mathbf{x}, \mathbf{z})$$

Since K_1 and K_2 are kernels, according to Mercer's condition, for every finite set $\{x_1, x_2, \ldots\}$, for any real valued choice of $c_1, c_2, \ldots$, we have

$$\sum_i \sum_j c_i c_j K_1(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \tag{1}$$

$$\sum_i \sum_j c_i c_j K_2(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \tag{2}$$

For $K_1(\mathbf{x},\mathbf{z})K_2(\mathbf{x},\mathbf{z})$ to be a Kernel, the following needs to be true

$$\sum_i \sum_j c_i c_j K_1(\mathbf{x}_i,\mathbf{x}_j) K_2(\mathbf{x}_i,\mathbf{x}_j) \geq 0 \tag{3}$$

Let $c_j K_1(\mathbf{x}_i,\mathbf{x}_j) = k_j$. Since c_j can take any real values, it follows that k_j can take any real value. So we can now reduce the requirement of the inequality 3 to

$$\sum_i \sum_j c_i k_j K_2(\mathbf{x}_i,\mathbf{x}_j) \geq 0 \tag{4}$$

However requirement 4 is equivalent to inequality 2, which is true. This means that $K_1(\mathbf{x},\mathbf{z})K_2(\mathbf{x},\mathbf{z})$ is a valid kernel.

(b) [10 points] Show that a polynomial over a kernel that is constructed using positive coefficients is a kernel. That is, if P is any polynomial with positive coefficients, show that K below is a kernel:

$$K(\mathbf{x},\mathbf{z}) = P(K_1(\mathbf{x},\mathbf{z}))$$

Hint: You may need show $K(\mathbf{x},\mathbf{z}) = \alpha K_1(\mathbf{x},\mathbf{z}) + \beta K_2(\mathbf{x},\mathbf{z})$ is a valid kernel and use the conclusion in the previous question.

We have already shown in the previous answer that the product of two kernels is also a kernel. That is, if $K_1(\mathbf{x},\mathbf{z})$ is a valid kernel, then $K_1(\mathbf{x},\mathbf{z})K_1(\mathbf{x},\mathbf{z})$. From this it follows that $(K_1(\mathbf{x},\mathbf{z}))^3$ is a kernel since $(K_1(\mathbf{x},\mathbf{z}))^3 = (K_1(\mathbf{x},\mathbf{z})K_1(\mathbf{x},\mathbf{z}))\,K_1(\mathbf{x},\mathbf{z}) = K_1'(\mathbf{x},\mathbf{z})K_1(\mathbf{x},\mathbf{z})$, where $K_1'(\mathbf{x},\mathbf{z}) = K_1(\mathbf{x},\mathbf{z})K_1(\mathbf{x},\mathbf{z})$. We already know that $K_1'(\mathbf{x},\mathbf{z})K_1(\mathbf{x},\mathbf{z})$ is a valid kernel since its the product of two kernels. This means that $(K_1(\mathbf{x},\mathbf{z}))^{n+1}$ is a kernel if $(K_1(\mathbf{x},\mathbf{z}))^n$ is a kernel. Using induction we can therefore show that $(K_1(\mathbf{x},\mathbf{z}))^n$ is a kernel for any whole number n.

A polynomial over a kernel that is constructed using positive coefficients can be shown as a combination of many terms of the form $\alpha K_1(\mathbf{x},\mathbf{z}) + \beta K_2(\mathbf{x},\mathbf{z}) + \ldots$. Using Mercer's condition, we need to show that

$$\sum_i \sum_j c_i k_j \left(\alpha K_1(\mathbf{x}_i,\mathbf{z}_j) + \beta K_2(\mathbf{x}_i,\mathbf{z}_j) + \ldots\right) \geq 0 \tag{5}$$

$$\alpha \sum_i \sum_j c_i k_j K_1(\mathbf{x}_i,\mathbf{z}_j) + \beta \sum_i \sum_j c_i k_j K_2(\mathbf{x}_i,\mathbf{z}_j) + \cdots \geq 0 \tag{6}$$

Since $K_1(\mathbf{x},\mathbf{z})$ and $K_2(\mathbf{x},\mathbf{z})$ and other terms are kernels, and coefficients α, β and others are positive, relation 6 is true. Which means that a polynomial over a kernel that is constructed using positive coefficients is a kernel.

2. [10 points] Given two examples $\mathbf{x} \in \Re^2$ and $\mathbf{z} \in \Re^2$, let

$$K(\mathbf{x}, \mathbf{z}) = 15 \left(\mathbf{x}^T \mathbf{z}\right)^2 \exp\left(-||\mathbf{x} - \mathbf{z}||^2\right) \tag{7}$$

Prove that this is a valid kernel function.

According to Mercer's condition, for every finite set $\{x_1, x_2, \ldots\}$, for any real valued choice of $c_1, c_2, \ldots$, the following needs to be true

$$\sum_i \sum_j c_i c_j 15 \left(\mathbf{x}_i^T \mathbf{z}_j\right)^2 \exp\left(-||\mathbf{x}_i - \mathbf{z}_j||^2\right) \geq 0$$

We know that $\exp\left(-||\mathbf{x}_i - \mathbf{z}_j||^2\right)$ will always be positive. Let $\exp\left(-||\mathbf{x}_i - \mathbf{z}_j||^2\right) = k_{ij}$. So the inequality that must be true is

$$15 k_{ij} \sum_i \sum_j c_i c_j \left(\mathbf{x}_i^T \mathbf{z}_j\right)^2 \geq 0$$

If we define $\left(\mathbf{x}_i^T \mathbf{z}_j\right)^2$ as $(K(\mathbf{x}_i, \mathbf{z}_j))^2$, then we can represent kernel $(K(\mathbf{x}_i, \mathbf{z}_j))^2$ as another kernel $K'(\mathbf{x}_i, \mathbf{z}_j)$ since we have already shown that any power of a kernel is also a kernel. We now have to show that

$$15 k_{ij} \sum_i \sum_j c_i c_j K'(\mathbf{x}_i, \mathbf{z}_j) \geq 0$$

which is true by Mercer's condition since $K'(\mathbf{x}_i, \mathbf{z}_j)$ is a kernel.

3. (**For 6350 students**)[10 points]An valid kernel can always be expressed as inner product. Prove that the Gaussian kernel

$$K(\mathbf{x}, \mathbf{z}) = \exp(\frac{-\|x - z\|^2}{2\sigma^2})$$

can be written down as the inner product of an feature space with infinite dimension. Hint: You may do some expansion and then show the middle factor can be expanded as a power series.

$$\begin{aligned}
K(\mathbf{x}, \mathbf{z}) &= \exp\left(\frac{-\|x - z\|^2}{2\sigma^2}\right) \\
&= \exp\left(\frac{-x^2 + 2xz - z^2}{2\sigma^2}\right) \\
&= \exp\left(\frac{-x^2}{2\sigma^2}\right) \exp\left(\frac{2xz}{2\sigma^2}\right) \exp\left(\frac{-z^2}{2\sigma^2}\right) \\
&= \exp\left(\frac{-x^2}{2\sigma^2}\right) \left(1 + \frac{\frac{xz}{\sigma^2}}{1!} + \frac{\left(\frac{xz}{\sigma^2}\right)^2}{2!} + \frac{\left(\frac{xz}{\sigma^2}\right)^3}{3!} + \ldots\right) \exp\left(\frac{-z^2}{2\sigma^2}\right) \\
&= \exp\left(\frac{-x^2}{2\sigma^2}\right) \left(1 + \frac{1}{\sqrt{1!}} \frac{x}{\sigma} \frac{1}{\sqrt{1!}} \frac{z}{\sigma} + \frac{1}{\sqrt{2!}} \left(\frac{x}{\sigma}\right)^2 \frac{1}{\sqrt{2!}} \left(\frac{z}{\sigma}\right)^2 + \ldots\right) \exp\left(\frac{-z^2}{2\sigma^2}\right)
\end{aligned}$$

This can be seen to be a product in a different dimension between

$$\phi(x) = \exp(\frac{-x^2}{2\sigma^2}) \left(1, \frac{x}{\sqrt{1!}\sigma}, \frac{x^2}{\sqrt{2!}\sigma^2}, \frac{x^3}{\sqrt{3!}\sigma^3}, \dots\right)$$

and

$$\phi(z) = \exp(\frac{-z^2}{2\sigma^2}) \left(1, \frac{z}{\sqrt{1!}\sigma}, \frac{z^2}{\sqrt{2!}\sigma^2}, \frac{z^3}{\sqrt{3!}\sigma^3}, \dots\right)$$

Which means $K(\mathbf{x}, \mathbf{z})$ can be written down as the inner product of a feature space with infinite dimensions.

3 Experiments

In this question, you will implement the support vector machine (SVM) and a variant of random forest which combine SVMs and decision trees.

We will use two datasets for this question:

1. `semelion handwritten digits data`: This dataset contains 1593 handwritten digits from around 80 persons were scanned, stretched in a rectangular box 16x16 in a gray scale of 256 values. Our goal is implement svm in this data set to determine whether is a number 6.

2. `madelon`: This is one of five datasets used in the NIPS 2003 feature selection challenge. There are 2000 examples in training set and 600 examples in test set.

You may reuse your code in decision tree. If **you have problems in decision tree**, please contact with TA to get help. You may use Java, Python, Matlab, C/C++ for this assignment. If you want to use a different language, you must contact the instructor first. Any other language you may want to use **MUST** run on the CADE machines.

3.1 Support Vector Machines

1. [6 points] Implement SVM in handwriting dataset with hyperparameter $C = 1$ and $\gamma_0 = 0.01$. Report the accuracy in test set and training set.

   Note:

   (a) update learning rate according to:

   $$\gamma_t = \frac{\gamma_0}{1 + \gamma_0 * t/c}$$

   in this question, as well as the following questions related to SVM.

   (b) Don't forget to add bias item as the first dimension of $\mathbf{x}$.

The accuracy ($\frac{TP+TN}{Number of Samples}$) obtained was

Data set	Accuracy
Test	0.7184
Training	0.7240

Table 3: Handwriting data set accuracy on test and training sets

2. [8 points] Run your SVM code on the `madelon` dataset and use 5-fold cross-validation to choose suitable parameters. At least attempt 6 different values for C and 3 different values for γ_0. Report the average accuracy for each group of parameters. Report the accuracy in your test set as well as training set.

   Hint: You should try out C in exponential steps, for example, $2^1, 2^{-1}, 2^{-2}, \cdots$.

Starting Learning Rate γ_0	Tradeoff Value C	Average Accuracy
10.0	2.0	0.4940
10.0	1.0	0.5885
10.0	0.5	0.5000
10.0	0.25	0.4880
10.0	0.125	0.5000
10.0	0.0625	0.5000
10.0	0.03125	0.5040
10.0	0.015625	0.4870
10.0	0.0078125	0.5000
10.0	0.00390625	0.5000
10.0	0.001953125	0.4980
10.0	$9.765625E-4$	0.5000
1.0	2.0	0.5035
1.0	1.0	0.5655
1.0	0.5	0.4900
1.0	0.25	0.5020
1.0	0.125	0.5000
1.0	0.0625	0.5000
1.0	0.03125	0.5000
1.0	0.015625	0.4980
1.0	0.0078125	0.5000
1.0	0.00390625	0.5050
1.0	0.001953125	0.5000
1.0	$9.765625E-4$	0.5000
0.1	2.0	0.5785
0.1	1.0	0.5130
0.1	0.5	0.5500
0.1	0.25	0.4495

0.1	0.125	0.5610
0.1	0.0625	0.5010
0.1	0.03125	0.5030
0.1	0.015625	0.5000
0.1	0.0078125	0.5000
0.1	0.00390625	0.4810
0.1	0.001953125	0.5010
0.1	$9.765625E-4$	0.5050
0.01	2.0	0.5285
0.01	1.0	0.5950
0.01	0.5	0.5105
0.01	0.25	0.4950
0.01	0.125	0.5695
0.01	0.0625	0.5315
0.01	0.03125	0.5245
0.01	0.015625	0.5575
0.01	0.0078125	0.5565
0.01	0.00390625	0.5000
0.01	0.001953125	0.5000
0.01	$9.765625E-4$	0.5000
0.001	2.0	0.5710
0.001	1.0	0.5630
0.001	0.5	0.5970
0.001	0.25	0.5775
0.001	0.125	0.5770
0.001	0.0625	0.5095
0.001	0.03125	0.4615
0.001	0.015625	0.4775
0.001	0.0078125	0.5545
0.001	0.00390625	0.5310
0.001	0.001953125	0.5000
0.001	$9.765625E-4$	0.5000
0.0001	2.0	0.5615
0.0001	1.0	0.5630
0.0001	0.5	0.5750
0.0001	0.25	0.5770
0.0001	0.125	0.5890
0.0001	0.0625	0.4650
0.0001	0.03125	0.5640
0.0001	0.015625	0.5620
0.0001	0.0078125	0.4875
0.0001	0.00390625	0.4940

0.0001	0.001953125	0.5000
0.0001	$9.765625E-4$	0.4820
0.00001	2.0	0.5525
0.00001	1.0	0.5880
0.00001	0.5	0.5820
0.00001	0.25	0.5955
0.00001	0.125	0.5965
0.00001	0.0625	0.5900
0.00001	0.03125	0.4960
0.00001	0.015625	0.5180
0.00001	0.0078125	0.4995
0.00001	0.00390625	0.5000
0.00001	0.001953125	0.5000
0.00001	$9.765625E-4$	0.5000
0.000001	2.0	0.5670
0.000001	1.0	0.5880
0.000001	0.5	0.6050
0.000001	0.25	0.5945
0.000001	0.125	0.5690
0.000001	0.0625	0.4830
0.000001	0.03125	0.4860
0.000001	0.015625	0.4860
0.000001	0.0078125	0.5000
0.000001	0.00390625	0.5000
0.000001	0.001953125	0.4860
0.000001	$9.765625E-4$	0.5000
0.0000001	2.0	0.5860
0.0000001	1.0	0.5480
0.0000001	0.5	0.5980
0.0000001	0.25	0.5850
0.0000001	0.125	0.4745
0.0000001	0.0625	0.4760
0.0000001	0.03125	0.4895
0.0000001	0.015625	0.4870
0.0000001	0.0078125	0.4820
0.0000001	0.00390625	0.5000
0.0000001	0.001953125	0.4990
0.0000001	$9.765625E-4$	0.4900
0.00000001	2.0	0.5420
0.00000001	1.0	0.5925
0.00000001	0.5	0.5465
0.00000001	0.25	0.4955

0.00000001	0.125	0.4810
0.00000001	0.0625	0.4710
0.00000001	0.03125	0.4810
0.00000001	0.015625	0.4870
0.00000001	0.0078125	0.4810
0.00000001	0.00390625	0.4910
0.00000001	0.001953125	0.5000
0.00000001	$9.765625E-4$	0.4930
0.000000001	2.0	0.5210
0.000000001	1.0	0.4920
0.000000001	0.5	0.5010
0.000000001	0.25	0.4880
0.000000001	0.125	0.4850
0.000000001	0.0625	0.4780
0.000000001	0.03125	0.4740
0.000000001	0.015625	0.4740
0.000000001	0.0078125	0.4810
0.000000001	0.00390625	0.4920
0.000000001	0.001953125	0.4750
0.000000001	$9.765625E-4$	0.4910
0.0000000001	2.0	0.5130
0.0000000001	1.0	0.4930
0.0000000001	0.5	0.4790
0.0000000001	0.25	0.4770
0.0000000001	0.125	0.4780
0.0000000001	0.0625	0.4710
0.0000000001	0.03125	0.4780
0.0000000001	0.015625	0.4790
0.0000000001	0.0078125	0.5065
0.0000000001	0.00390625	0.4750
0.0000000001	0.001953125	0.4850
0.0000000001	$9.765625E-4$	0.4870

Table 4: Madelon data set average accuracy for hyper parameters

Due to the extensive cross validation used, the run took around 7 to 8 minutes. The above table was obtained for the SVM run log, that is written into a text file. The above table was obtained by copying data written into file NonEnsembleMadelonLogFile.txt by the SVM program.

Data set	Accuracy
Test	0.5867
Training	0.6325

Table 5: Madelon data set accuracy on test and training sets

All metrics are written into log file SvmRunLogFile.txt and have been copied into the table above.

3. [6 points] Precision , recall and F_1 score are another metrics besides accuracy, which are useful if the dataset is unbalanced with respect to the positive and negative examples. To compute these quantities, you should count the number of true positives (that is, examples that your classifier predicts as positive and are truly positive), the false positives (i.e, examples that your classifier predicts as positive, but are actually labeled negative) and the false negatives (i.e., examples that are predicted as negative by your classifier, but are actually positive).

   Denote true positives, false positive and false negative as TP, FP and FN respectively. The precision (p), recall (r) and f-value F_1 are defined as:

$$\begin{aligned} p &= \frac{TP}{TP + FP} \\ r &= \frac{TP}{TP + FN} \\ F_1 &= 2\frac{p \cdot r}{p + r} \end{aligned}$$

   Give precision, recall and F_1 score for your classifiers constructed in the previous two questions.

Data set	Precision	Recall	F_1 Score
Handwriting test	0.6476	0.9967	0.7851
Handwriting training	0.6650	1.0000	0.7988
Madelon test	0.5802	0.6267	0.6026
Madelon training	0.6233	0.6700	0.6458

Table 6: Precision, Recall and F_1 score for classifiers run on Handwriting and Madelon data sets

All metrics are written into log file SvmRunLogFile.txt and have been copied into the tables above.

3.2 Ensemble of decision trees

Recall that a random forest is an ensemble based on bagging and decision tree. For bagging, we draw m samples *with replacement* from the training set. According to

$$\lim_{m\to\infty} (1 - \frac{1}{m})^m \to \frac{1}{e} \simeq 0.368$$

there are about 63.2 percent items may not appear that set. In random forest, we use this sampling method N times, to construct N training sets and grow N decision trees. Note that we build unpruned decision trees.

In each node, instead of using the best feature by ID3 , we choose k features randomly and than use the ID3 heuristic to find the best feature to split on. Generally, $k = log_2 d$ is a good choice where d is the number of features for our data.

Since there are N trees, there will be N predictions for each example. Generally, the final prediction is voted on by these trees. However, we would like to use SVM to combine these predictions for this question. Specifically, after growing the N decision trees, you should construct a new D consisting of transformed features. The feature transformation $\phi(x)$ is defined using the N trees as follows:

$$\phi(x) = [tree_1(x), tree_2(x), \cdots, tree_N(x)]$$

In other words, you will build an N dimensional vector consisting of the prediction (1 or -1) of each tree that you created. Thus, you have a *learned* feature transformation.

You will finally train an SVM on this new dataset D.

1. [15 points] Using the method mentioned above, construct $N = 5$ decision trees for the `handwriting` dataset. For each node, select $k = log_2 d = 8$ features randomly and then use the ID3 heuristic to find the best feature for splitting.

   Train the SVM meta-classifier and report the accuracy for both training set and test set. (No cross-validation is required but please choose good parameters for SVM, we will take out points for very low accuracy.)

   Extensive cross validation was used. The parameters were the same as those used for SVM classification (without the forest) on the madelon dataset.

Data set	Accuracy	Precision	Recall	F_1 Score
Test	0.6897	0.7460	0.6046	0.6679
Training	0.7800	0.8475	0.7299	0.7843

Table 7: Accuracy, Precision, Recall and F_1 score for classifier run on Handwriting data set with forest of 5 trees

2. [25 points] Implement same method on the `madelon` dataset. ($k = log_2 d = 11$)

   (a) [20 points] Try $N = 10, 30, 100$. For each N, report accuracy on training set and test set. (cross-validation is not required, but please choose good parameters for the SVM.)

   Extensive cross validation was used. The parameters were the same as those used for SVM classification (without the forest) on the madelon dataset.

Data set	Number of Trees	Accuracy
Test	10	0.5533
Test	30	0.5650
Test	100	0.5467
Training	10	0.6285
Training	30	0.7450
Training	100	0.8350

Table 8: Accuracy by number of trees for classifier run on Madelon data set

(b) [5 points] Choose the best N among those you have tried (you may try some new numbers). Report the accuracy, precision, recall and f_1 score on test set.

The F_1 score was best with 10 trees and the accuracy was best with 30 trees.

Best number of trees	Accuracy	Precision	Recall	F_1 Score
10	0.5533	0.5408	0.7067	0.6127
30	0.5650	0.5644	0.5700	0.5672

Table 9: Accuracy, Precision, Recall and F_1 score for classifier run on Madelon test data set

All metrics are written into log file SvmRunLogFile.txt and have been copied into the tables above.