67577 | Introduction to Machine Learning | Exercise 3

Guy Lutsker 207029448

Question 1

Bayes Optimal and LDA

Consider binary classification with sample space $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \{\pm 1\}$. One way to model the data generation process is to assume that our samples are drawn i.i.d from an unknown **joint** distribution $\mathcal{D}$ over $\mathcal{X} \times \{\pm 1\}$. (Namely, we draw the sample and the label together from a joint distribution over $\mathcal{X} \times \{\pm 1\}$.) In this question you'll learn about two concepts that were not discussed in the lecture: The Bayes Optimal Classifier, and Linear Discriminent Analysis (LDA).

1. If we knew $\mathcal{D}$, our best predictor would have been assigning the class with the higher probability:

$$\forall \mathbf{x} \in \mathcal{X}\ h_{\mathcal{D}}(\mathbf{x}) = \begin{cases} +1 & \Pr(y=1|\mathbf{x}) \geqslant \frac{1}{2} \\ -1 & otherwise \end{cases}$$

where the probability is over $\mathcal{D}$. This classifier is known as the **Bayes Optimal** classifier.

Show that

$$h_{\mathcal{D}} = \underset{y \in \{\pm 1\}}{\operatorname{argmax}} \Pr(\mathbf{x}|y) \Pr(y).$$

Solution :

We know that Bayes rule says that $P(X|Y) \cdot P(Y) = P(Y|X) \cdot P(X)$ and so,

$h_{\mathcal{D}} = \begin{cases} +1 & P(y=1|x) \geq \frac{1}{2} \\ -1 & else \end{cases}$ we can take a look at $P(y=1|x) \overset{Bayes}{=} \frac{P(x|y=1) \cdot P(y=1)}{P(x)}$

if the function returns 1 that means that $P(y=1|x) \geq 0.5$ which means:

$1 = h_{\mathcal{D}}(x) \rightarrow P(y=1|x) \geq 0.5 \rightarrow 1 = \underset{y \in \{\pm 1\}}{argmax}\{P(y=1|x)\} \overset{Bayes}{=} \underset{y \in \{\pm 1\}}{argmax}\{\frac{P(x|y=1) \cdot P(y=1)}{P(x)}\}$

$\overset{doesnt\ change\ argmax}{=} \underset{y \in \{\pm 1\}}{argmax}\{P(x|y=1) \cdot P(y=1)\}$

the same it true for $-1 = h_{\mathcal{D}}(x)$ and so over all we get that $h_{\mathcal{D}} = \underset{y \in \{\pm 1\}}{argmax}\{P(x|y) \cdot P(y)\}$

Question 2

2. Assume that $\mathcal{X} = \mathbb{R}^d$ and that $\mathbf{x}|y \sim \mathcal{N}(\mu_y, \Sigma)$ for some mean vector $\mu_y \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$ (that is, the covariance matrix Σ is the same for both $y \in \{\pm 1\}$, but the expectation μ_y is different for each $y \in \{\pm 1\}$). In other words,

$$f(\mathbf{x}|y) = \frac{1}{\sqrt{(2\pi)^d \det(\Sigma)}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu_y)^\top \Sigma^{-1}(\mathbf{x}-\mu_y)\right\}$$

where f is the density function for the multivariate normal distribution. Show that in this case, if we knew μ_{+1}, μ_{-1} and Σ then the Bayes Optimal classifier is

$$h_{\mathcal{D}}(\mathbf{x}) = \operatorname*{argmax}_{y\in\{\pm 1\}} \delta_y(\mathbf{x}),$$

where δ_{+1} and δ_{-1} are functions $\mathbb{R}^d \to \mathbb{R}$ given by

$$\delta_y(x) = \mathbf{x}^\top \Sigma^{-1}\mu_y - \frac{1}{2}\mu_y^\top \Sigma^{-1}\mu_y + \ln \Pr(y) \qquad y \in \{\pm 1\}$$

The functions $\delta_{\pm 1}$ are called **discriminant functions**: this classification rule predicts the label, based on which of the two discriminant functions is larger at the sample $\mathbf{x}$ we wish to classify.

Solution :

We have shown in the previous question that $h_{\mathcal{D}} = \underset{y\in\{\pm1\}}{argmax}\{P(x|y)\cdot P(y)\}$

here we get that $P = f$ and that we are looking at the multidimensional case where we know the density function. then lets look at:

$h_{\mathcal{D}} = \underset{y\in\{\pm1\}}{argmax}\{P(x|y)\cdot P(y)\} \overset{P=f}{=} \underset{y\in\{\pm1\}}{argmax}\{f(x|y)\cdot P(y)\}=$

$\underset{y\in\{\pm1\}}{argmax}\{f(x|y)\cdot P(y)\} = \underset{y\in\{\pm1\}}{argmax}\{\frac{1}{\sqrt{(2\pi)^d det(\Sigma)}}\text{Exp}(-\frac{1}{2}(x-\mu_y)^\intercal\Sigma^{-1}(x-\mu_y))\cdot P(y)\}$

$\overset{doesnt\ change\ argmax}{=} \underset{y\in\{\pm1\}}{argmax}\{\text{Exp}(-\frac{1}{2}(x-\mu_y)^\intercal\Sigma^{-1}(x-\mu_y))\cdot P(y)\}$

$\overset{\text{Exp } is\ monotonian}{=} \underset{y\in\{\pm1\}}{argmax}\{-\frac{1}{2}(x-\mu_y)^\intercal\Sigma^{-1}(x-\mu_y) + ln(P(y))\}$

$= \underset{y\in\{\pm1\}}{argmax}\{(-\frac{1}{2}x^\intercal + \frac{1}{2}\mu_y^\intercal)\cdot\Sigma^{-1}(x-\mu_y) + ln(P(y))\}$

$= \underset{y\in\{\pm1\}}{argmax}\{(-\frac{1}{2}x^\intercal\Sigma^{-1} + \frac{1}{2}\mu_y^\intercal\Sigma^{-1})\cdot(x-\mu_y) + ln(P(y))\}$

$= \underset{y\in\{\pm1\}}{argmax}\{-\frac{1}{2}x^\intercal\Sigma^{-1}x + \frac{1}{2}\mu_y^\intercal\Sigma^{-1}x + \frac{1}{2}x^\intercal\Sigma^{-1}\mu_y - \frac{1}{2}\mu_y^\intercal\Sigma^{-1}\mu_y + ln(P(y))\}$

$= \underset{y\in\{\pm1\}}{argmax}\{x^\intercal\Sigma^{-1}\mu_y - \frac{1}{2}\mu_y^\intercal\Sigma^{-1}\mu_y + ln(P(y))\}$

$= \underset{y\in\{\pm1\}}{argmax}\{\delta_y(x)\}$

Question 3

3. In practice, we don't know μ_{+1}, μ_{-1}, Σ and $\Pr(y)$. In order to turn the above into a classifier, given a training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, we need to estimate them. Write your formula for estimating μ_{+1}, μ_{-1}, Σ and $\Pr(y)$ based on S.

   When you plug these estimates into the functions $\delta_{\pm 1}$, you get a classifier known as **Linear Discriminant Analysis (LDA)**.

   for all sets of point that their y=1, say there were $x_1, \ldots x_n$ such points
   we will approximate thusly:
   $\mu_{+1} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and in a similar matter $\mu_{-1} = \frac{1}{m-n}\sum_{j=1}^{m-n} x_j$
   as to Σ we can approximate it like this:
   $\Sigma = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \mu_i)(x_i - \mu_i)^\intercal$
   and $P(y=1) = \frac{\sum_{i=1}^{m}\delta(y_i,1)}{m}$, $P(y=0) = \frac{\sum_{i=1}^{m}\delta(y_i,0)}{m}$ where δ is Kronecker delta.

Question 4

Spam

4. You are building a spam filter - a classifier that receives an email and decides whether it's a spam message or not. What are the two kinds of errors that your classifier could make? Which of them is the error we really don't want to make? Which of the labels $\{\text{spam}, \text{not-spam}\}$ should be the **negative** label and which should be the **positive** label, if we want the false-positive error (Type-I error) to be the error we really don't want to make?

   We can make two types of mistakes:
   - False positive: classifying a non-spam email as spam.
   - False negative: classifying a spam email as non-spam email.
   The first error is worse since this way we can miss an important email.

Question 5

SVM- Formulation

5. The canonical form of a Quadratic Program (QP) is:

$$\underset{\mathbf{v}\in\mathbb{R}^n}{\text{argmin}}\left(\frac{1}{2}\mathbf{v}^\top Q\mathbf{v}+\mathbf{a}^\top\mathbf{v}\right)$$
$$\text{s.t. } A\mathbf{v}\leqslant\mathbf{d},$$

where $Q\in\mathbb{R}^{n\times n}, A\in\mathbb{R}^{m\times n}, \mathbf{a}\in\mathbb{R}^n, \mathbf{d}\in\mathbb{R}^m$ are fixed vectors and matrices.

Write the Hard-SVM problem as a QP problem in canonical form. Specifically, using the Hard-SVM problem formulation

$$\underset{(\mathbf{w},b)}{\text{argmin}}\,||\mathbf{w}||^2 \text{ s.t. } \forall i, y_i(\langle\mathbf{w},\mathbf{x}_i\rangle+b)\geqslant 1.$$

what are the values of $Q, A, \mathbf{a}, \mathbf{d}$ that express this problem as a QP in canonical form?

Why is it interesting? QP software solvers take QP in canonical form. To use a QP solver, you'll need to express the SVM problem as QP in canonical form as above.

Solution :

We've got the Hard-SVM problem: $\underset{w,b}{argmin}\{\|w\| \,|\forall i,\ y_i(\langle w,x_i\rangle+b)\geq 1\}$
we need to find $Q\in\mathbb{R}^{n\times n}, A\in\mathbb{R}^{m\times n}, a\in\mathbb{R}^n, d\in\mathbb{R}^m$
to convert it into the problem $\underset{v\in\mathbb{R}^n}{argmin}\{\frac{1}{2}v^\intercal Qv+a^\intercal v\,|Av\leq d\}$.
Well, firstly we can choose $Q=I_n$ and $a=v\cdot\frac{1}{2}$ and we will get:
$\frac{1}{2}v^\intercal Qv+a^\intercal v=\frac{1}{2}v^\intercal I_n v+\frac{1}{2}v^\intercal\cdot v=v^\intercal v=\|v\|^2$
Then we need to convert $y_i(\langle w,x_i\rangle+b)\geq 1$ into $Av\leq d$
We can describe it as $\langle w,x_i\rangle\geq\frac{1-b\cdot y_i}{y_i}$
and in matrix nation: $A\in\mathbb{R}^{m\times n}$,let $A_i$be the i'th row of A.
then denote $A_i=-x_i,1$ s.t $A=-\begin{bmatrix}x_1^\intercal,1\\ \vdots\\ x_m^\intercal,1\end{bmatrix}$ and $v=(w,b)$ and $d\in\mathbb{R}^m$ s.t $d_i=-\frac{1}{y_i}$

Question 6

6. In the Soft-SVM we defined the problem:

$$\arg\min_{\mathbf{w},\{\xi_i\}} \frac{\lambda}{2}||\mathbf{w}||^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i \text{ s.t. } \forall_i, y_i\langle \mathbf{w}, \mathbf{x}_i\rangle \geqslant 1-\xi_i \text{ and } \xi_i \geqslant 0$$

Show that this problem is equivalent to the problem (namely that these problem have the same solutions)

$$\arg\min_{\mathbf{w}} \frac{\lambda}{2}||\mathbf{w}||^2 + \frac{1}{m}\sum_{i=1}^{m}\ell^{hinge}(y_i\langle \mathbf{w}, \mathbf{x}_i\rangle),$$

where $\ell^{hinge}(a) = \max\{0, 1-a\}$.

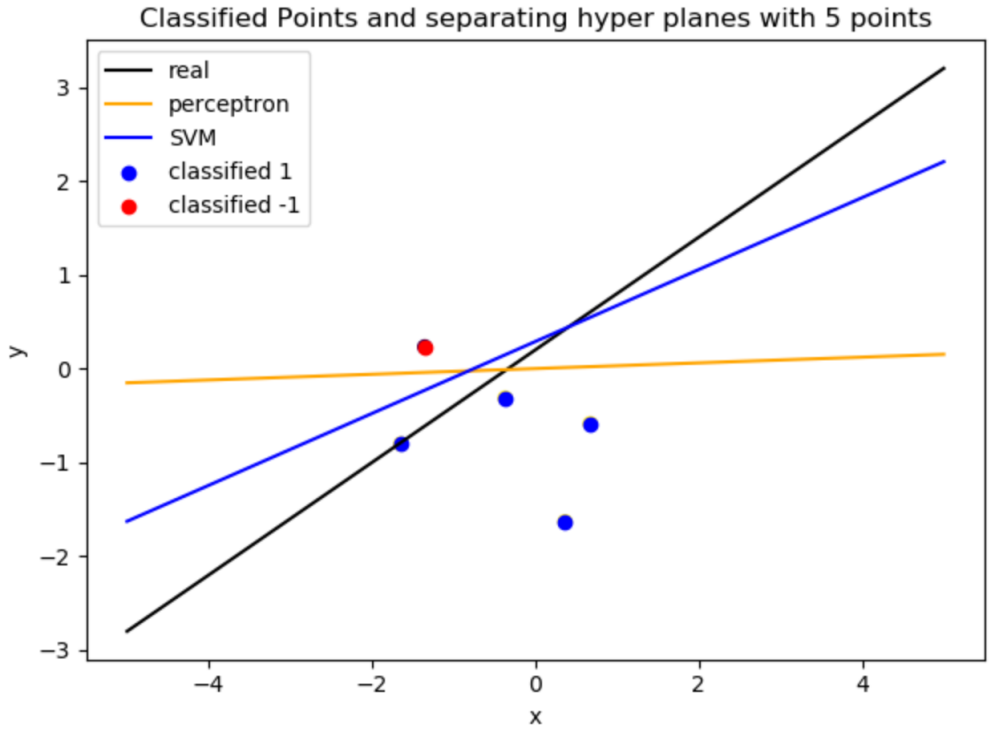
Solution :

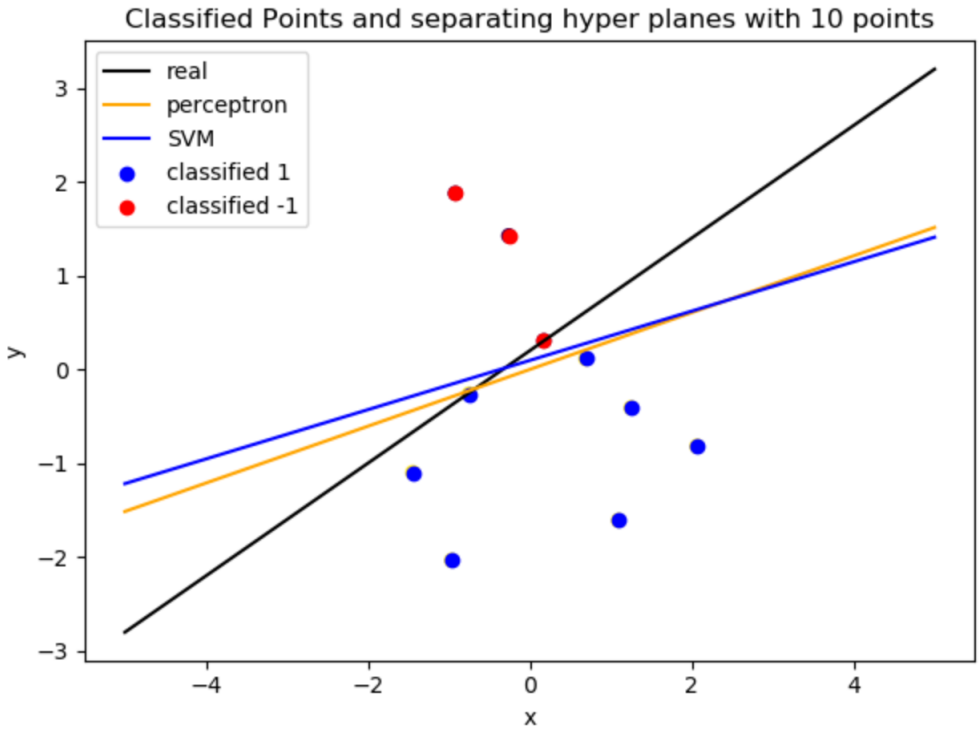
let there be some w
we would like for each $\xi_i$to be minimal, so that their average would be minimized.
and we also know that a legal solution must uphold $\xi_i \geq 1 - y_i\langle w, x_i\rangle \rightarrow \xi_i \geq l^{hinge}(y_i\langle w, x_i\rangle)$
we will show that indeed for ξ_i to be minimal it has to equal exactly $l^{hinge}(y_i\langle w, x_i\rangle)$.
if we will assume towards a contradiction that we have a solution $\xi_1 ... \xi_m$ such that
there exists i such that $\xi_i > l^{hinge}(y_i\langle w, x_i\rangle)$
then we can assign $\psi = l^{hinge}(y_i\langle w, x_i\rangle)$ and we shall notice that a solution that contains
ψ instead if ξ_i is still legal since $\psi \geq 0 \psi \geq 1 - y_i\langle w, x_i\rangle$.
and so we can construct a solution with $\xi_1 ... \xi_{i-1}, \psi, \xi_{i+1} ...$ that is lesser then the proposed minimal solution.
and so we can conclude that $\forall i\ \xi_i = l^{hinge}(y_i\langle w, x_i\rangle)$
as required:)

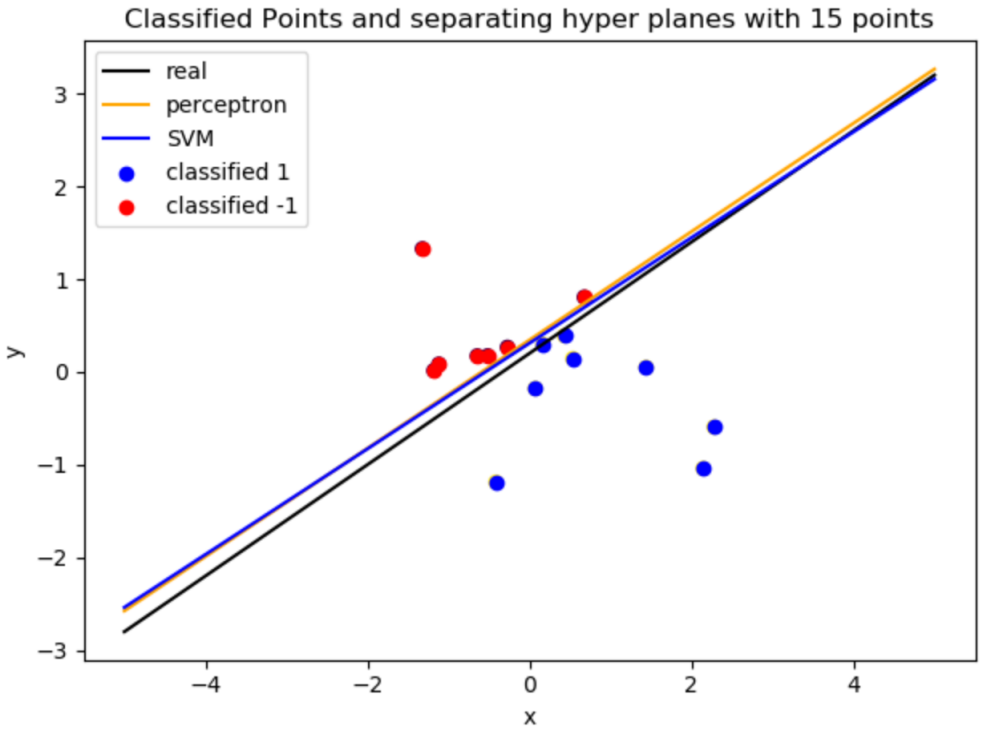
Coding Part :

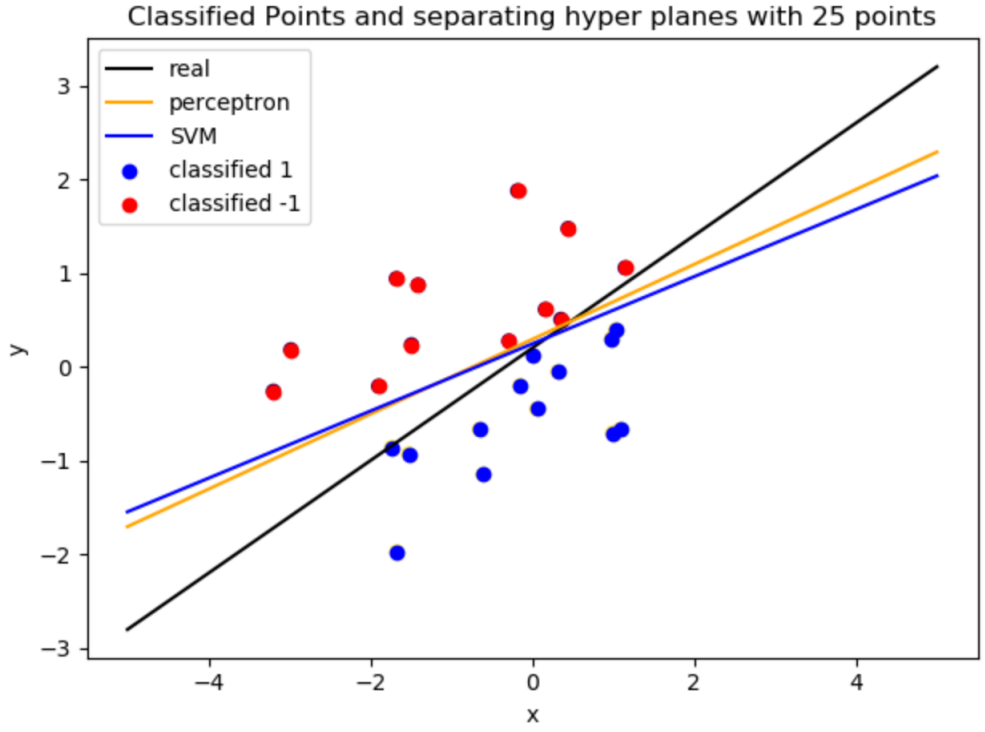
Question 9

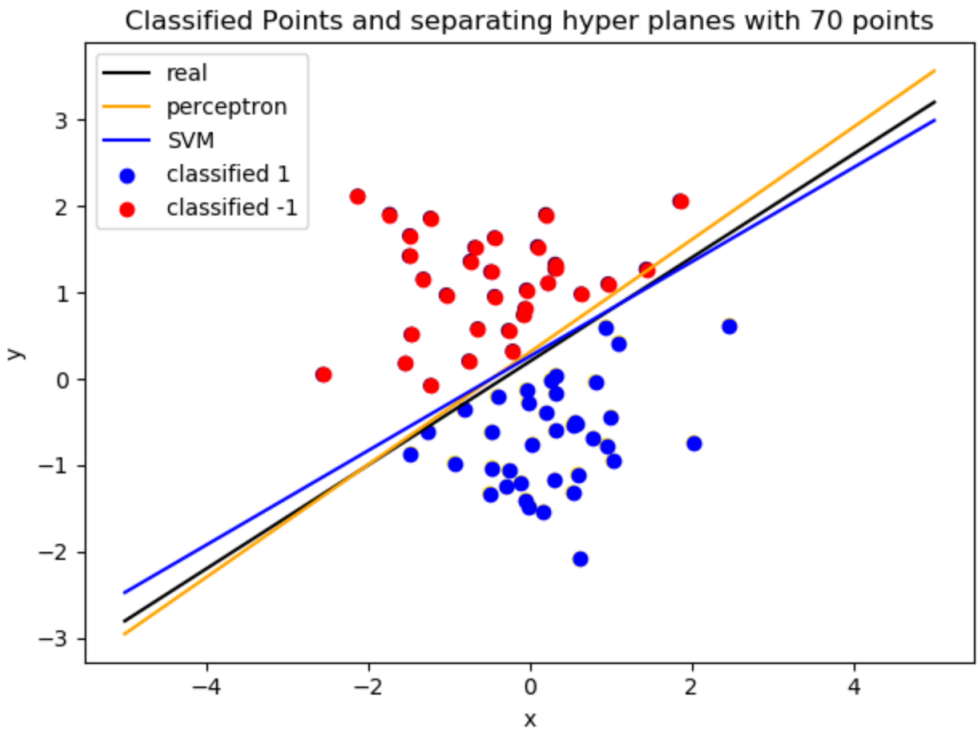
The five plots:





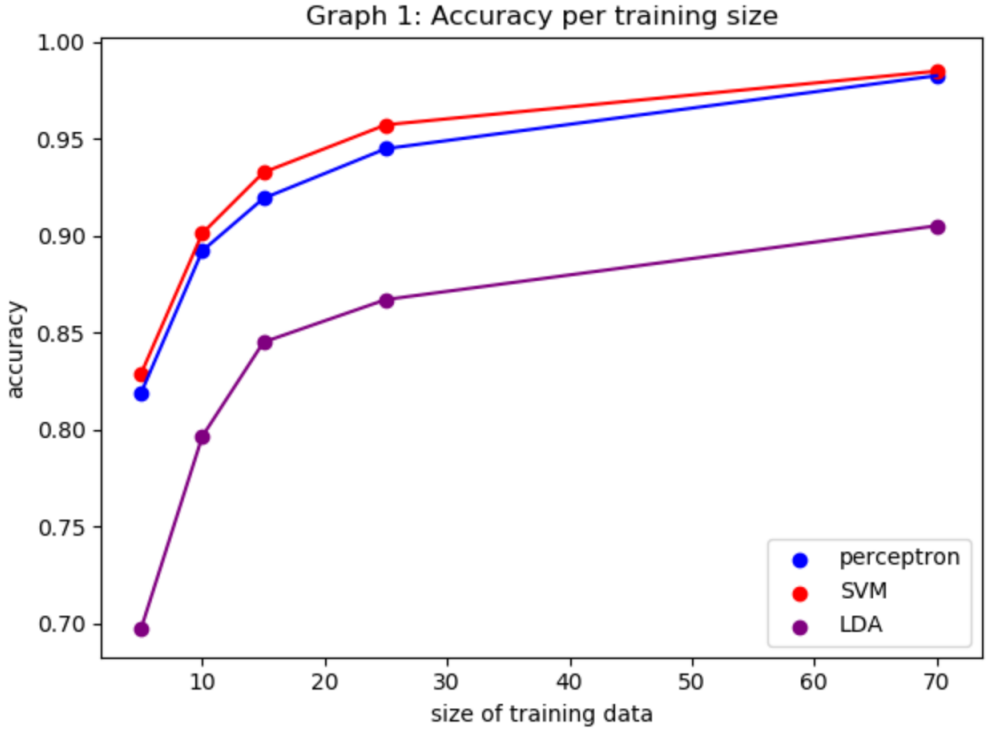

Classified Points and separating hyper planes with 15 points
real
perceptron
SVM
classified 1
classified -1
x
y


Classified Points and separating hyper planes with 25 points
real
perceptron
SVM
classified 1
classified -1
x
y



Question 10

The requested plot:



Question 11

Which classifier did better? why do you think that happened? No need for a formal argument, just explain what are the properties of the classifiers that cause these results.

Answer :

The best performing model was the SVM model, followed closely by the Perception model, and lastly the LDA model.

I believe that SVM performed the best because it tries to find the largest margin, and so provides another level of analysis that the other model don't.

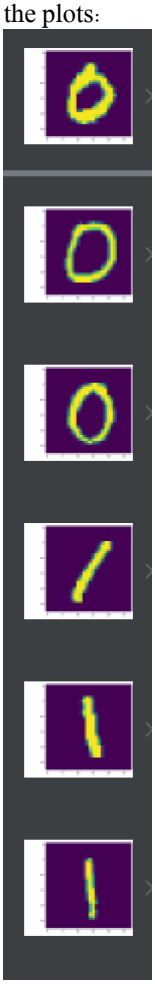
The perception also had accurate results(relatively) and it seems that this model fits the point pressingly and because

in this case they were linearly separable it provided a good hyperplane.

The LDA model did the worse in this run, i think this happens because it uses approximations in its formula and so could be a bit off.

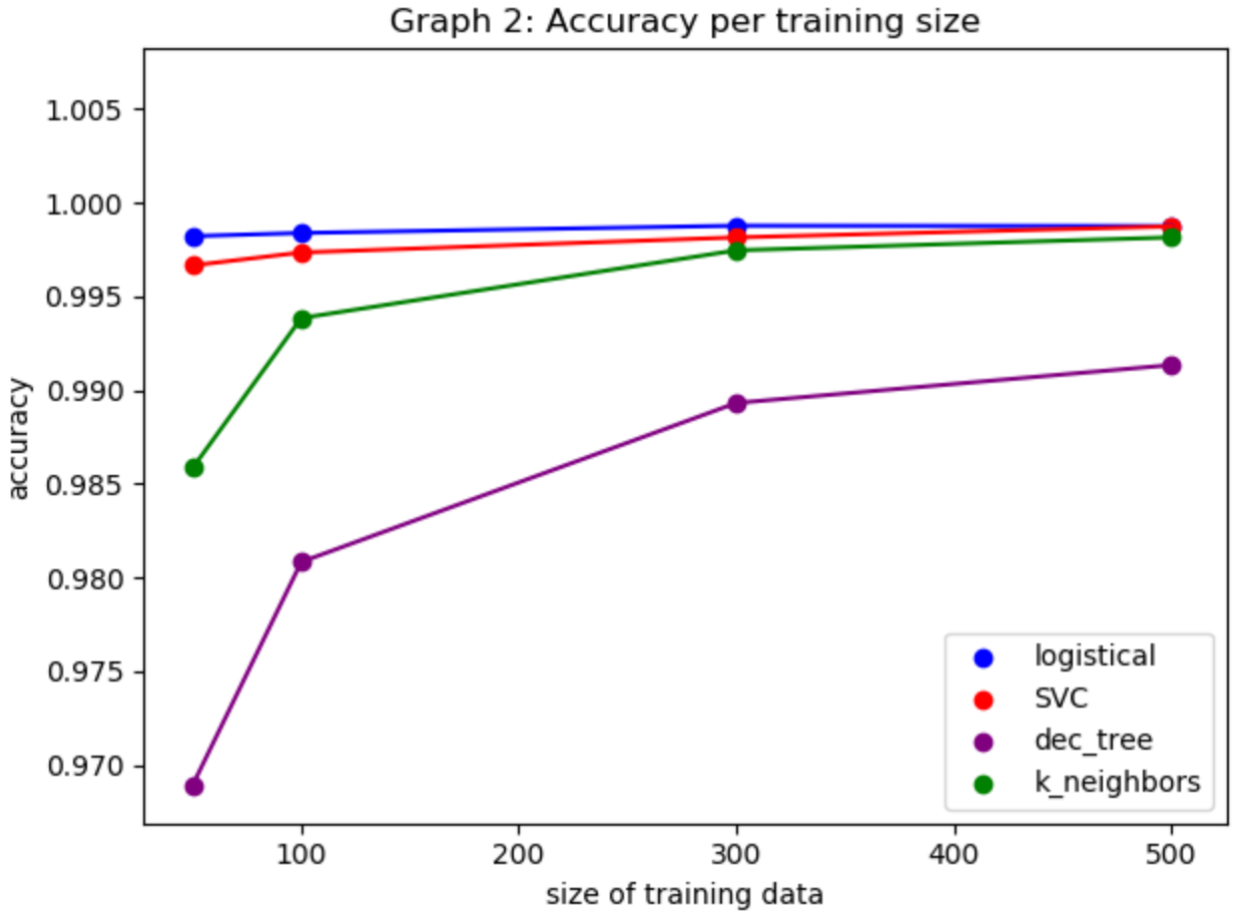
Question 12

the plots:



Question 14

the requested graph:



timing in ms as required:
logistical_time: 0.9626526832580566 on m: 50
svc_time: 4.530053377151489 on m: 50
dec_tree_time: 0.5432815551757812 on m: 50
k_neighbors_time: 12.160544872283936 on m: 50

logistical_time: 0.9805383682250977 on m: 100
svc_time: 5.738248109817505 on m: 100
dec_tree_time: 0.5887489318847656 on m: 100
k_neighbors_time: 19.974066734313965 on m: 100

logistical_time: 1.3659155368804932 on m: 300
svc_time: 8.841833114624023 on m: 300
dec_tree_time: 0.9401607513427734 on m: 300
k_neighbors_time: 56.468133211135864 on m: 300

logistical_time: 1.7132360935211182 on m: 500
svc_time: 11.144995927810669 on m: 500
dec_tree_time: 1.1772339344024658 on m: 500
k_neighbors_time: 95.177410364151 on m: 500