Introduction to Machine Learning (67577)
Exercise III – Classification
Hadar Sharvit – 208287599

Theoretical Questions

Bayes Optimal and LDA

1. Show that $h_D = \underset{y\in\{\pm1\}}{\operatorname{argmax}}(\Pr(\mathrm{x}|\mathrm{y})\Pr(y))$:
We know from bayes theorem that $P(x|y)P(y) = P(y|x)$, therefore it is sufficient to show that $h_D = \underset{y\in\{\pm1\}}{\operatorname{argmax}}\big(P(y|x)\big) = \underset{y}{\operatorname{argmax}}(\{P(y=1|x), P(y=-1|x)\})$.
Suppose in the first case that $\Pr(y=1|x) \geq 1/2$. If so, $\Pr(y=-1|x) < 1/2$ thus the corresponding argmax would be $y=+1$ (as $\Pr(y=1|x) > \Pr(y=-1|x)$). For the other case, $\Pr(y=1|x) < 1/2$, which in turn means $\Pr(y=-1|x) \geq 1/2$, so argmax would be $y=-1$. We can write those in compart form as followed
$$\underset{y\in\{\pm1\}}{\operatorname{argmax}}(\Pr(\mathrm{x}|\mathrm{y})\Pr(y)) = \underset{y\in\{\pm1\}}{\operatorname{argmax}}(\mathrm{P}(\mathrm{y}|\mathrm{x})) = \left\{\begin{matrix} +1, & \Pr(y=1|x) \geq 1/2 \\ -1, & otherwise \end{matrix}\right\} = h_D(x)$$

2. Show that $h_D(x) = \underset{y\in\{\pm1\}}{\operatorname{argmax}}\ \delta_y(x)$ where $\delta_y(x) = x^T\Sigma^{-1}\mu_y - \frac{1}{2}\mu_y^T\Sigma^{-1}\mu_y + \ln\Pr(y)$:
From Q.1 we have $h_D(x) = \underset{y\in\{\pm1\}}{\operatorname{argmax}}(\Pr(\mathrm{x}|\mathrm{y})\Pr(y))$, and we know that for calculating argmax, taking $\log h_D$ will provide the same result. If so, let us concentrate on the following:
$$\ln\Pr(\mathrm{x}|\mathrm{y})\Pr(y) = \ln\Pr(x|y) + \ln\Pr(y)$$
Substituting $f(x|y)$ we have
$$\begin{aligned}\ln\Pr(x|y) &= \ln\left(\frac{1}{\sqrt{(2\pi)^d\det\Sigma}}\exp\left(-\frac{1}{2}(x-\mu_y)^T\Sigma^{-1}(x-\mu_y)\right)\right)\\ &= -\ln\left(\frac{1}{\sqrt{(2\pi)^d\,det\,\Sigma}}\right) - \frac{1}{2}(x-\mu_y)^T\Sigma^{-1}(x-\mu_y)\end{aligned}$$
Since the first ln is not a function of y, it will not affect the result of argmax. Let us simplify the second component
$$\begin{aligned}(x-\mu_y)^T\Sigma^{-1}(x-\mu_y) &= (x-\mu_y)^T(\Sigma^{-1}x - \Sigma^{-1}\mu_y) = (x^T-\mu_y^T)(\Sigma^{-1}x - \Sigma^{-1}\mu_y)\\ &= x^T\Sigma^{-1}x - x^T\Sigma^{-1}\mu_y - \mu_y^T\Sigma^{-1}x + \mu_y^T\Sigma^{-1}\mu_y\end{aligned}$$
Since Σ is the same for both ±1, $x^T\Sigma^{-1}\mu_y = \mu_y^T\Sigma^{-1}x$:
$$= x^T\Sigma^{-1}x - 2x^T\Sigma^{-1}\mu_y + \mu_y^T\Sigma^{-1}\mu_y$$
Yet again, $x^T\Sigma^{-1}x$ is not a function of y, therefore is not relevant for the calculation of argmax. In conclusion we have
$$\underset{\mathrm{y}\in\{\pm1\}}{\operatorname{argmax}}(\ln P(x|y)) = x^T\Sigma^{-1}\mu_y - \frac{1}{2}\mu_y^T\Sigma^{-1}\mu_y$$
And consequently
$$h_D(x) = \underset{y\in\{\pm1\}}{\operatorname{argmax}}(\Pr(\mathrm{x}|\mathrm{y})\Pr(y)) = \underset{y\in\{\pm1\}}{\operatorname{argmax}}\left(x^T\Sigma^{-1}\mu_y - \frac{1}{2}\mu_y^T\Sigma^{-1}\mu_y + \ln\Pr(y)\right) = \underset{y\in\{\pm1\}}{\operatorname{argmax}}\ \delta_y(x)$$

3. Write your formula for estimating $\mu_{\pm 1}$, Σ and $\Pr(y)$ based on $S = \{(x_i, y_i)\}_{i=1}^m$:
We can calculate the mean by summing the ratio of occurrences:
$$\mu_{\pm 1} = \text{mean}(x[y = \pm 1]) = \frac{\sum x_i[y = \pm 1]}{\sum 1[y = \pm 1]}$$
The probability would be the sum of occurrences divided by the total size
$$\Pr(y) = \frac{1}{m} \sum 1[y_i = y]$$
Similar to what we have seen in lecture 1, we can write
$$\Sigma_{\pm} = \text{cov}(\text{x}) = \frac{1}{m-1} \sum_{y = \pm 1} (x[y = \pm 1] - \widehat{\mu_{\pm}})(x[y = \pm 1] - \widehat{\mu_{\pm}})^T$$
And $\Sigma = \Sigma_+ + \Sigma_-$

Type I errors

4. Let us write the possible cases, given that $y = 1$ indicates spam and $y = -1$ is non-spam:
   1. If the current mail is non-spam ($y = -1$):
      a. $\hat{y} = -1$ (true negative): I have correctly identified the mail as non spam
      b. $\hat{y} = 1$ (false positive): I have declared the mail is spam (which is wrong)
   2. If the current mail is spam ($y = 1$):
      a. $\hat{y} = 1$ (true positive): I have correctly identified the mail as spam
      b. $\hat{y} = -1$(false negative): I have declared the mail is non-spam (which is wrong)

   1.b is denoted a Type-I error – we would wish to avoid declaring regular mail as spam

   2.b is not as bad, though still an error.

SVM – Formulation

5. Write the Hard-SVM problem as a QP problem:
We can set $Q = 2\mathbb{I}_n, a = \vec{0}_n$, so the QP takes the form of
$$\underset{\mathrm{v} \in \mathbb{R}^\mathrm{n}}{\text{argmin}}\ v^T v$$
$$s.t\ Av \leq d$$
Since $||w|| \geq 0$, $\text{argmax}\ ||\text{w}||^2 = \text{argmax}\ ||\text{w}||$, so for $v := w$ we have the QP argmin set. (as $v^T v = ||w||$)
Simplifying the conditions:
$$y_i \langle w, x_i \rangle + y_i b \geq 1 \rightarrow \langle w, x_i \rangle \geq \frac{1}{y_i} - b \overset{(*)}{=} y_i - b$$
$$-\langle w, x_i \rangle \leq b - y_i$$
Or in matrix form
$$= -\begin{pmatrix} w_1 & \cdots & w_n \end{pmatrix} \begin{pmatrix} | \\ x_i \\ | \end{pmatrix} \leq \begin{pmatrix} b - y_1 \\ \vdots \\ b - y_n \end{pmatrix}$$
$$= -\begin{pmatrix} - & x_i & - \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix} \leq \begin{pmatrix} b - y_1 \\ \vdots \\ b - y_n \end{pmatrix}$$

Therefore, in matrix form for every i:

$$-\underbrace{\begin{pmatrix} - & x_1 & - \\ \vdots & \vdots & \vdots \\ - & x_m & - \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix}}_{v} \leq \underbrace{\begin{pmatrix} b - y_1 \\ \vdots \\ b - y_n \end{pmatrix}}_{d}$$

This finishes the transition to QP

$(*)$ $y_i \in \{\pm 1\} \to 1/y_i = y_i$

6. Showing problem equivalence

$$\underset{\substack{w,\xi_i \\ \forall i \ y_i\langle w,x_i\rangle \geq 1-\xi_i \\ \xi_i \geq 0}}{\operatorname{argmin}} \quad \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i =$$

Minimizing ξ_i separately

$$\underset{\substack{w \\ \forall i \ y_i\langle w,x_i\rangle \geq 1-\xi_i}}{\operatorname{argmin}} \quad \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m} \min_{\xi_i \geq 0} \xi_i =$$

Explicitly writing the condition for $\xi_i \geq 0$

$$= \underset{w}{\operatorname{argmin}} \frac{\lambda}{2}||w||^2 + \frac{1}{m} \cdot \begin{Bmatrix} \sum_{i=1}^{m} [1 - y_i\langle w, x_i\rangle], & 1 - y_i\langle w, x_i\rangle \geq 0 \\ 0, & else \end{Bmatrix} =$$

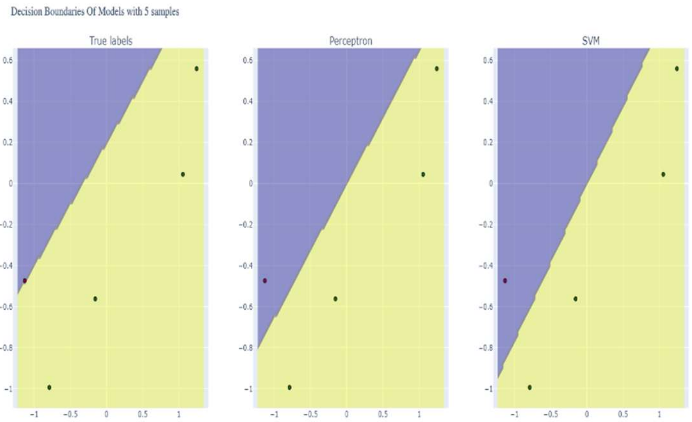
$$= \underset{w}{\operatorname{argmin}} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m} \max\{0{,}1 - y_i\langle w, x_i\rangle\}$$

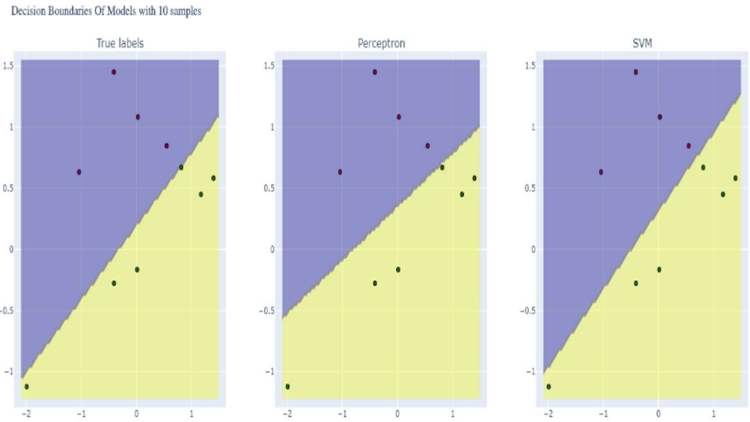
$$= \underset{w}{\operatorname{argmin}} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m} \ell^{hing} \ (y_i\langle w, x_i\rangle)$$

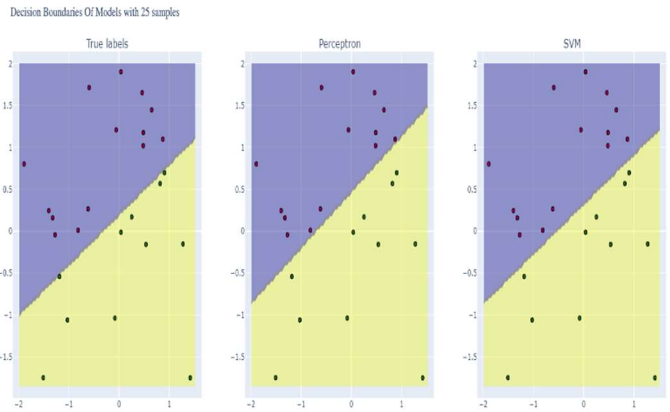
Practical Questions

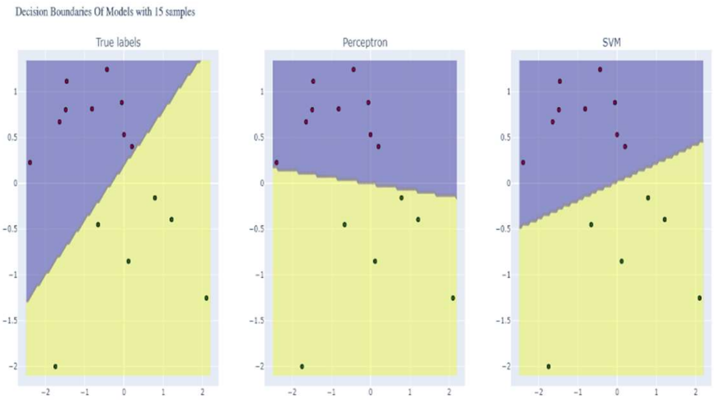
Implemention and simulation-comparison of different classifiers

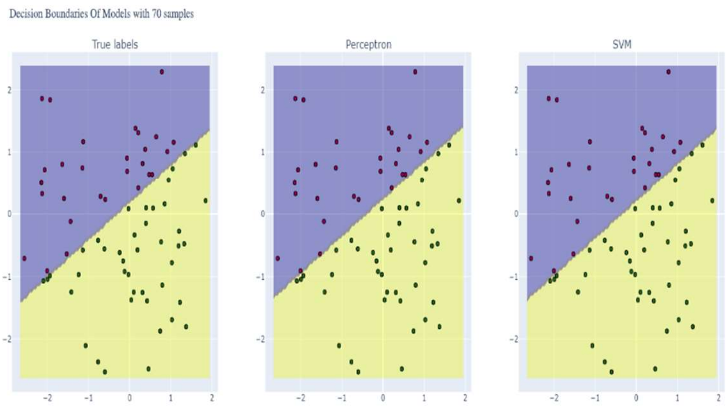
7. Code
8. Code
9. decision boundaries of models with m samples of True Labels, Perceptron and SVM



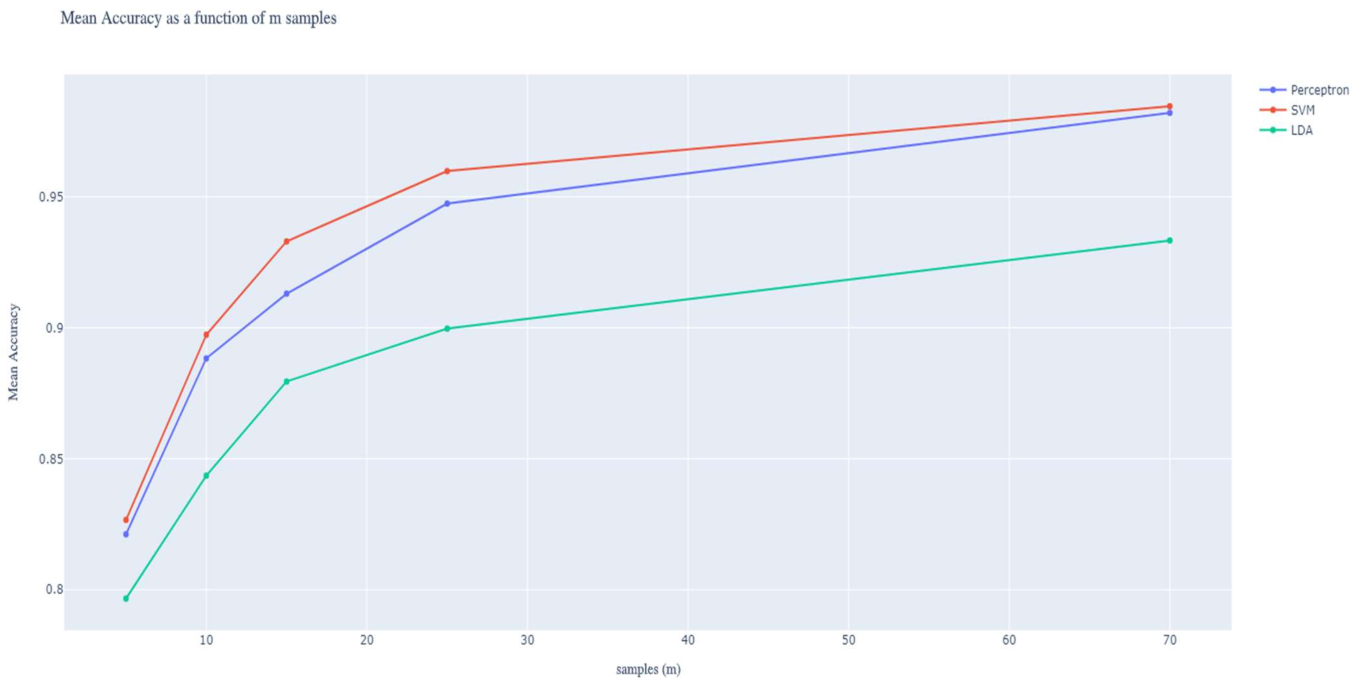








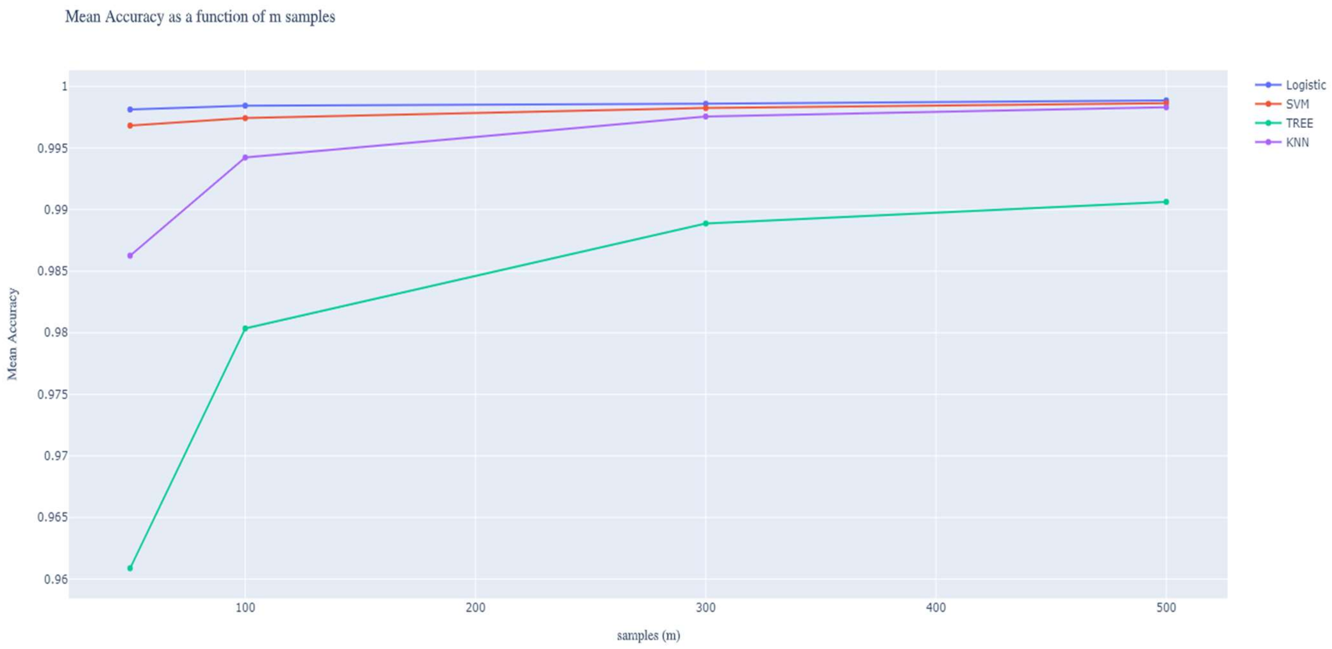
10. Mean accuracy as a function of m for SVM, Perceptron and LDA



11. It appears that classification using LDA returns lower accuracy rates. This might be due to the fact that LDA classification does not necessarily represent a plane, and this may cause overfitting. On the other hand, both Perceptron and SVM classification returned similar accuracies, and classified the data with high rates.
12. Code
13. Code

Classification of two digits from the MNIST Dataset

14. Mean accuracy as a function of m for Logistic Regression, Soft-SVM, Decision tree and K-Nearest Neighbors: we can see that all classifiers are able to classify with extremely high accuracy, though Decision tree handles the classification with a little less (though still good) accuracy rates. This might be due to the fact that I have set the maximum depth rather low.



We could also plot the mean time t as a function of m, in the following plot, it is clear that the Decision Tree classifier takes the longest to classify