Introduction to Machine Learning (67577)

Exercise 3 - Classification

Dor Roter
208772251

May 12, 2021

Contents

1 Bayes Optimal and LDA

1. let

$$\forall x \in \mathcal{X} h_{\mathcal{D}}(x) = \begin{cases} +1 & \mathbb{P}\left(y=1|x\right) \geq \frac{1}{2} \\ -1 & otherwise \end{cases}$$

Using Bayes equation it holds that for every $x \in \mathcal{X}$ and $y \in \mathcal{Y}$:

$$\underset{y\in\{\pm 1\}}{argmax} \mathbb{P}\left(x|y\right)\mathbb{P}\left(y\right) \underset{Bayes}{=} \underset{y\in\{\pm 1\}}{argmax} \mathbb{P}\left(x,y\right) \underset{Bayes}{=} \underset{y\in\{\pm 1\}}{argmax} \mathbb{P}\left(y|x\right)\mathbb{P}\left(x\right) = \begin{cases} +1 & \mathbb{P}\left(y=1|x\right)\mathbb{P}\left(x\right) \geq \mathbb{P}\left(y=-1|x\right)\mathbb{P}\left(x\right) \\ -1 & otherwise \end{cases}$$

now, as $y \in \mathcal{Y} = \{\pm 1\}$:

$$\forall x \in \mathcal{X} \quad \mathbb{P}\left(y=1|x\right) + \mathbb{P}\left(y=-1|x\right) = 1$$

$$\begin{aligned} &\Rightarrow \mathbb{P}\left(y=1|x\right)\mathbb{P}\left(x\right) \geq \mathbb{P}\left(y=-1|x\right)\mathbb{P}\left(x\right) \\ &\Leftrightarrow \mathbb{P}\left(y=1|x\right) \geq \mathbb{P}\left(y=-1|x\right) \\ &\Leftrightarrow \mathbb{P}\left(y=1|x\right) \geq \frac{1}{2} \end{aligned}$$

And therefore it holds as required that:

$$\forall x \in \mathcal{X} \quad \underset{y\in\{\pm 1\}}{argmax} \mathbb{P}\left(x|y\right)\mathbb{P}\left(y\right) = h_{\mathcal{D}}(x) = \begin{cases} +1 & \mathbb{P}\left(y=1|x\right) \geq \frac{1}{2} \\ -1 & otherwise \end{cases} \qquad \square$$

2. Let us note:

$$h_{\mathcal{D}}\left(x\right) = \underset{y\in\{\pm 1\}}{argmax} \mathbb{P}\left(x|y\right)\mathbb{P}\left(y\right) = \underset{y\in\{\pm 1\}}{argmax} \mathbb{P}\left(x|y\right)\mathbb{P}\left(y\right) = \underset{y\in\{\pm 1\}}{argmax}\, \overline{\delta}_y(x)$$

where $\overline{\delta}_y(x) = \mathbb{P}\left(x|y\right)\mathbb{P}\left(y\right)$.
Therefore, since the ln function is a monotonically increasing function, finding the maximizer $y \in \{\pm 1\}$ for $\overline{\delta}$ is in fact equivalent to finding the maximizer over:

$$\overline{\delta}'_y(x) = ln\left(\mathbb{P}\left(x|y\right)\mathbb{P}\left(y\right)\right)$$

And thus:

$$\begin{aligned} &\overline{\delta}'_y(x) = ln\left(\mathbb{P}\left(x|y\right)\mathbb{P}\left(y\right)\right) = ln\left(f(x|y)\mathbb{P}(y)\right) = \\ &ln\left(\frac{1}{\sqrt{\left(2\pi\right)^d det\left(\Sigma\right)}} \cdot exp\left\{-\frac{1}{2}\left(x-\mu_y\right)^T \Sigma^{-1}\left(x-\mu_y\right)\right\}\mathbb{P}(y)\right) = \\ &ln\left(exp\left\{-\frac{1}{2}\left(x-\mu_y\right)^T \Sigma^{-1}\left(x-\mu_y\right)\right\}\right) - ln\left(\sqrt{\left(2\pi\right)^d det\left(\Sigma\right)}\right) + ln\left(\mathbb{P}(y)\right) = \\ &-\frac{1}{2}\left(x^T - \mu_y^T\right)\left(\Sigma^{-1}x - \Sigma^{-1}\mu_y\right) - ln\left(\sqrt{\left(2\pi\right)^d det\left(\Sigma\right)}\right) + ln\left(\mathbb{P}(y)\right) \end{aligned}$$

Since d, x and Σ are constants relative to y, finding the maximizer to $\overline{\delta}'$ is also equivalent to finding the maximizing y over:

$$\begin{aligned}
&\delta'_y(x) = \overline{\delta}'_y(x) + ln\left(\sqrt{(2\pi)^d\, det\,(\Sigma)}\right) = \\
&\quad -\frac{1}{2}\left(x^T - \mu_y^T\right)\left(\Sigma^{-1}x - \Sigma^{-1}\mu_y\right) + ln\left(\mathbb{P}(y)\right) = \\
&\quad \frac{1}{2}\left(\left(x^T\Sigma^{-1}\mu_y - \mu_y^T\Sigma^{-1}\mu_y\right) + \left(\mu_y^T\Sigma^{-1}x - x^T\Sigma^{-1}x\right)\right) + ln\left(\mathbb{P}(y)\right)
\end{aligned}$$

once again, as $x^T\Sigma^{-1}x$ is constant in relation to y, maximizing $\overline{\delta}$' over y is equivalent to finding the maximizer y over:

$$\begin{aligned}
&\delta_y(x) = \delta'_y(x) + \frac{1}{2}x^T\Sigma^{-1}x = \\
&\frac{1}{2}\left(\left(x^T\Sigma^{-1}\mu_y - \mu_y^T\Sigma^{-1}\mu_y\right) + \left(\mu_y^T\Sigma^{-1}x - x^T\Sigma^{-1}x\right)\right) + ln\left(\mathbb{P}(y)\right) = \\
&\frac{1}{2}\left(x^T\Sigma^{-1}\mu_y + \mu_y^T\Sigma^{-1}x\right) - \frac{1}{2}\mu_y^T\Sigma^{-1}\mu_y + ln\left(\mathbb{P}(y)\right)
\end{aligned}$$

now we note that:

$$\begin{aligned}
&x^T\Sigma^{-1}\mu_y + \mu_y^T\Sigma^{-1}x = x^T\Sigma^{-1}\mu_y + \left\langle \mu_y | \Sigma^{-1}x \right\rangle = \\
&x^T\Sigma^{-1}\mu_y + \left\langle \Sigma^{-1}x | \mu_y \right\rangle = \\
&x^T\Sigma^{-1}\mu_y + \left(\Sigma^{-1}x\right)^T\mu_y = \\
&x^T\Sigma^{-1}\mu_y + x^T\left(\Sigma^{-1}\right)^T\mu_y
\end{aligned}$$

since Σ is the covariance matrix, its a symetric $d \times d$ matrix:

$$\Sigma^T = \Sigma \Rightarrow \Sigma^{-1} = \underbrace{\left(\Sigma^T\right)^{-1} = \left(\Sigma^{-1}\right)^T}_{\text{transpose properties}}$$

and thus $\Sigma^{-1} = \left(\Sigma^{-1}\right)^T$ and it holds that:

$$x^T\Sigma^{-1}\mu_y + \mu_y^T\Sigma^{-1}x = x^T\Sigma^{-1}\mu_y + x^T\left(\Sigma^{-1}\right)^T\mu_y = 2x^T\Sigma^{-1}\mu_y$$

$$\begin{aligned}
\Rightarrow \delta_y(x) = \quad & \frac{1}{2}\left(x^T\Sigma^{-1}\mu_y + \mu_y^T\Sigma^{-1}x\right) - \frac{1}{2}\mu_y^T\Sigma^{-1}\mu_y + ln\left(\mathbb{P}(y)\right) = \\
& x^T\Sigma^{-1}\mu_y - \frac{1}{2}\mu_y^T\Sigma^{-1}\mu_y + ln\left(\mathbb{P}(y)\right)
\end{aligned}$$

summing it all up we get

$$h_{\mathcal{D}}(x) = \underset{y\in\{\pm1\}}{argmax}\,\mathbb{P}\left(x|y\right)\mathbb{P}\left(y\right) = \underset{y\in\{\pm1\}}{argmax}\ \delta_y(x) \qquad \square$$

3. As we have seen in statistics and in recitation the unbiased estimator for mean is the average:

$$\hat{\mu} = \frac{1}{|S|}\sum_{x_i \in S} x_i$$

since we aim to compute the mean for each of the classes separately, we need to compute it seperately for each class and thus we can denote it as follows using indicator notation:

$$\forall y \in \{\pm 1\} \quad \hat{\mu}_y = \frac{1}{\sum_{i=1}^{m} \mathbb{1}\left[y_i = y\right]} \sum_{i=1}^{m} x_i \cdot \mathbb{1}\left[y_i = y\right]$$

Next, we can use our knowledge of the mean estimator to compute the covariance matrix unbiased estimator ($\Sigma = \mathbb{E}\left[\left(X - \mathbb{E}\left[X\right]\right)\left(X - \mathbb{E}\left[X\right]\right)^T\right]$).
Furthermore, we note that for each sample class we have a different mean (μ_y) and thus the covariance matrix for all samples can be commputed by seperating the computation of each $y \in \{\pm 1\}$ and using the appropriate μ_y:

$$\overline{\Sigma} = \frac{1}{m} \sum_{y \in \{\pm 1\}} \sum_{i \in \{j \in [m] | y_j = y\}} \left[\left(x_i - \hat{\mu_y}\right)\left(x_i - \hat{\mu_y}\right)^T\right]$$

By computing the bias we can note that this estimator is biased by a factor of $\frac{m}{m-1}$, and thus an unbiased estimator for the covariance matrix would be:

$$\hat{\Sigma} = \frac{1}{m-1} \sum_{y \in \{\pm 1\}} \sum_{i \in \{j \in [m] | y_j = y\}} \left[\left(x_i - \hat{\mu_y}\right)\left(x_i - \hat{\mu_y}\right)^T\right]$$

Lastly, as seen in the recitiation:

$$\hat{p}_{MLE} = \frac{1}{m}\mathbb{1}\left[y_i = 1\right]$$

2 Spam

4. The two error kinds our classifier might make tagging a normal email as spam, or not tagging a spam email as spam.
Out of the two error types, we are more likely to prefer our classifier to be a bit “looser”, allowing some spam to pass through, rather than too strict, which could lead the classifier to classifying important emails as spam which might lead our user to miss them.
Accordingly, we would note the tag “spam” as the posetive label, and the tag “not-spam” as the negative label, and thuse our Type-I (false-posetive) error would be tagging an email as spam, when it’s not (which we have decided to be the worse error out of the two).

3 SVM- Formulation

5. First let us note that $||w||^2 = \langle w|w\rangle$ and thus, by denoting

$$Q = 2\begin{bmatrix} I_d & \\ & 0 \end{bmatrix}$$
$$v = \begin{pmatrix} w \\ b \end{pmatrix}$$
$$a = 0_{d+1}$$

we achieve $||w||^2 = \langle w|w\rangle = w^T w = \frac{1}{2}w^T 2I_d w + \underbrace{0^T v = \frac{1}{2}v^T}_{\text{block matrix mult.}} Qv + a^T v$.

next, we will define A and d by constructing it so that each row $A_i v \le d_i$ is equivalent to the equation $y_i\left(\langle w|x_i\rangle + b\right) \ge 1$:

$$\begin{aligned} & y_i\left(\langle w|x_i\rangle + b\right) = y_i\left(\langle x_i|w\rangle + b\right) = \\ & = y_i x_i^T w + y_i b = \begin{pmatrix} \cdots y_i x_i^T \cdots & y_i \end{pmatrix}\begin{pmatrix} w \\ b \end{pmatrix} \end{aligned}$$

$$\begin{aligned} & \Rightarrow y_i\left(\langle w|x_i\rangle + b\right) \ge 1 \Leftrightarrow -y_i\left(\langle w|x_i\rangle + b\right) \le -1 \\ & \Leftrightarrow -\begin{pmatrix} \cdots y_i x_i^T \cdots & y_i \end{pmatrix}\begin{pmatrix} w \\ b \end{pmatrix} \le -1 \end{aligned}$$

Therefore for $A_i = -\begin{pmatrix} \cdots y_i x_i^T \cdots & y_i \end{pmatrix}$ and $d_i = -1$ we achieve our goal of representing the constraints using the required notation:

$$A = -\begin{pmatrix} \cdots y_1 x_1^T \cdots & y_1 \\ \vdots & \vdots \\ \cdots y_m x_m^T \cdots & y_m \end{pmatrix}$$
$$d = \begin{pmatrix} -1 \\ \vdots \\ -1 \end{pmatrix}$$

6. Let us denote the following:

$$(*) \underset{w,\{\xi_i\}}{argmin} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i \text{ s.t } \forall i \; y_i\langle w|x_i\rangle \ge 1 - \xi_i \text{ and } \xi_i \ge 0$$
$$(**) \underset{w}{argmin} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m}\ell^{hinge}(y_i\langle w|x_i\rangle)$$

Since the minimization in the Soft-SVM problem is under the the constraints $y_i\langle w|x_i\rangle \ge 1 - \xi_i$ and $\xi_i \ge 0$

for each $i \in [m]$, let $w \in \mathbb{R}^d$ it holds that:

$$\begin{aligned}&\forall i \in [m]\ y_i \langle w|x_i\rangle \geq 1 - \xi_i \ \wedge\ \xi_i \geq 0\\ &\Leftrightarrow \xi_i \geq 1 - y_i \langle w|x_i\rangle \ \wedge\ \xi_i \geq 0\\ &\Leftrightarrow \xi_i \geq max\left\{1 - y_i \langle w|x_i\rangle, 0\right\} = \ell^{hinge}(y_i \langle w|x_i\rangle)\end{aligned}$$

Therefore, for any $w \in \mathbb{R}^d$ it holds that the best ξ_i for the porpuse of minimization under the provided constraints is $\underline{\boldsymbol{\xi_i = \ell^{hinge}(y_i \langle w|x_i\rangle)}}$, as any smaller value would not be legal by the lema above.

Therefore, assuming by way of contradiction that there exists $w' \in \mathbb{R}^d$ such as that w' is a solution of $(*)$ but is not a solution of $(**)$ we get:

$$\begin{aligned}&\exists \{\xi_i\}\ s.t\ (w', \{\xi_i'\}) \in (*)\\ &\Rightarrow \underset{w,\{\xi_i\}}{min} \left\{\frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i \,|\forall i\ y_i \langle w|x_i\rangle \geq 1 - \xi_i \text{ and } \xi_i \geq 0\right\} =\\ &\frac{\lambda}{2}||w'||^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i' \underset{\text{prev. lema}}{\geq} \frac{\lambda}{2}||w'||^2 + \frac{1}{m}\sum_{i=1}^{m} \ell^{hinge}(y_i \langle w'|x_i\rangle)\end{aligned}$$

since, w' is **not** a solution of $(**)$, it must not minimize the expression, and thus for any $w \in \mathbb{R}^d$, where w is a solution of $(**)$ it holds that:

$$\begin{aligned}\frac{\lambda}{2}||w'||^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i' \underset{\text{prev. lema}}{=} \frac{\lambda}{2}||w'||^2 + \frac{1}{m}\sum_{i=1}^{m} \ell^{hinge}(y_i \langle w'|x_i\rangle) \underset{assumption}{>} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m} \ell^{hinge}(y_i \langle w|x_i\rangle)\\ = \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i\end{aligned}$$

And thus any the solution $w', \{\xi_i'\}$ is not a minimizer of the expression, and we reach a contradiction to the initial assumption.
Accordingly, for any $w \in \mathbb{R}^d$ such as that w is a solution of $(**)$ but not a solution of $(*)$, we reach by using similar claims to a contradiction to w being a minimizer:

$$\begin{aligned}\frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m} \ell^{hinge}(y_i \langle w|x_i\rangle) \underset{\text{prev. lema}}{=} \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i > \frac{\lambda}{2}||w'||^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i' =\\ \underset{\text{prev. lema}}{=} \frac{\lambda}{2}||w'||^2 + \frac{1}{m}\sum_{i=1}^{m} \ell^{hinge}(y_i \langle w'|x_i\rangle)\end{aligned}$$

Proving that in fact:

$$\begin{aligned}&(*)\underset{w,\{\xi_i\}}{argmin}\ \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i \text{ s.t } \forall i\ y_i \langle w|x_i\rangle \geq 1 - \xi_i \text{ and } \xi_i \geq 0\\ &\Updownarrow \qquad \square\\ &(**)\underset{w}{argmin}\ \frac{\lambda}{2}||w||^2 + \frac{1}{m}\sum_{i=1}^{m} \ell^{hinge}(y_i \langle w|x_i\rangle)\end{aligned}$$

4 Implemention and simulation-comparison of different classifiers

9.

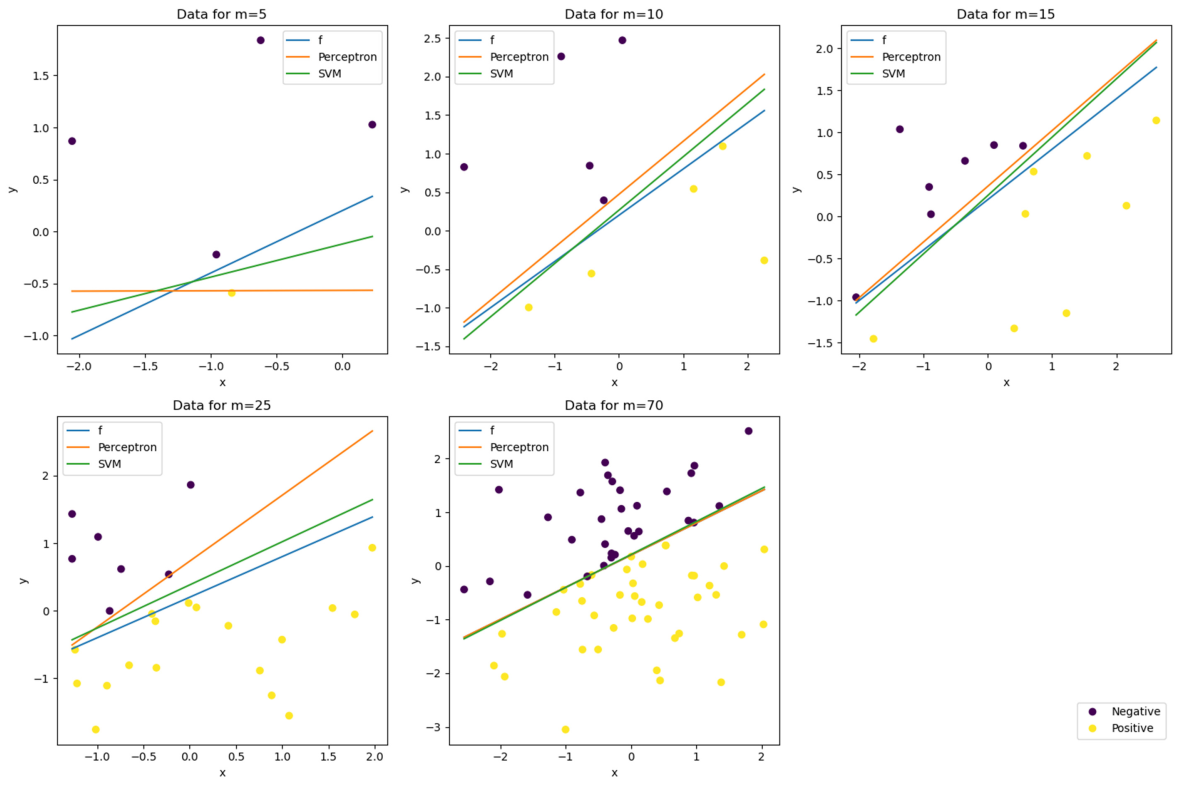

Figure 1: Data plots with classifiers hypothesis hyperplanes

10.

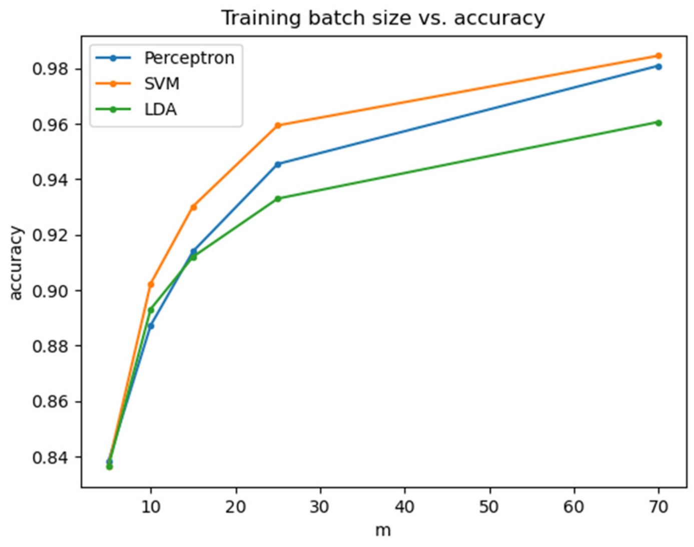

Figure 2: Training set size effect on the classifiers average performance

11. It is clear that the SVM classifier outpreformed the rest of the classifiers, with the Perceptron algorithm being right behind, and the LDA coming at last.
    First thing first, as SVM maximizes the margin, it seems quit logical that it generelizes well. Next, the Perceptron optimizes by "small" steps over each sample which might be why it came so close to the SVM.
    Lastly, the LDA model assumes the data was sampled from two gaussians sharing the same covariance matrix, since the data was not created this way, it's not surprising the LDA model did not preform as well.

5 MNIST Dataset

14. As expected k-nearest-neighbors has the highest running time as it just stores the training set, and uses it to go through the entirety of the data in order to make each prediction which makes it relatively expensive computationaly speaking.
    Next, we have the Soft-SVM algorithm, which as we have seen needs to calcualted the support vectors by testing each sample against all others and thus its also effected by the increase in dataset size as seen in the plot below.
    So clearly those 2 models training/prediction is affected by the training set size as they require several operations to be made for each sample.
    Lastly, both decisions trees and logistic regresion are quit fast as both are able to make a prediction and train using constant amount of operations on the samples (I have limited the tree's hight and thus the training set size does not effect the running time as much).

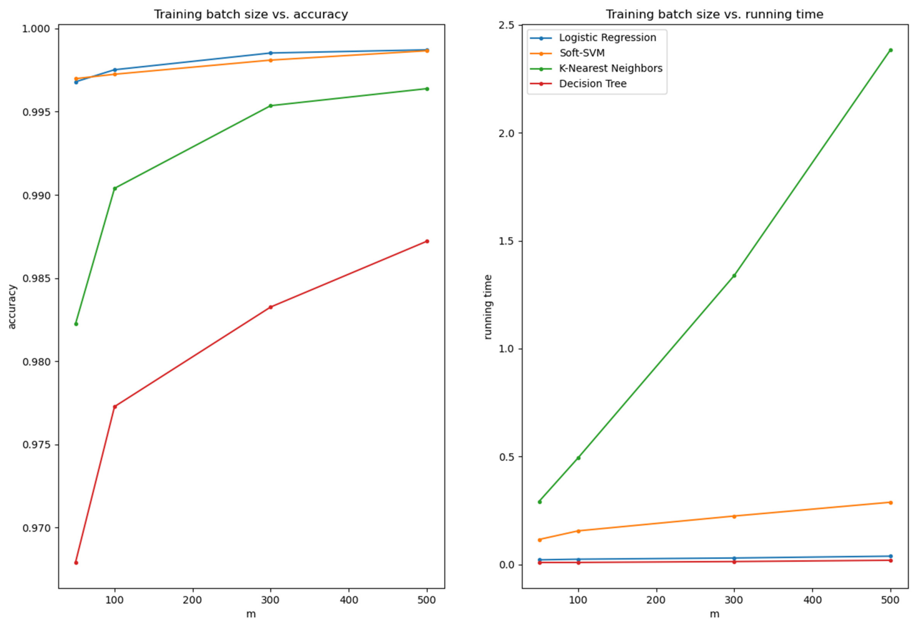

Figure 3: Classification models accuracy and running times against rising training set sizes